RQF LEVEL 5

GENPP501

NETWORKING AND INTERNET TECHNOLOGIES

Python Programming Fundamentals

TRAINER'S MANUAL

October, 2024

PYTHON PROGRAMMING FUNDAMENTALS

2024

AUTHOR'S NOTE PAGE (COPYRIGHT)

© **Rwanda TVET Board**

Copies available from:

- *HQs: Rwanda TVET Board-RTB*
- *Web: www.rtb.gov.rw*
- **KIGALI-RWANDA**

Original published version: October, 2024

ACKNOWLEDGEMENTS

The publisher would like to thank the following for their assistance in the elaboration of this training manual:

Rwanda TVET Board (RTB) extends its appreciation to all parties who contributed to the development of the trainer and trainee’s manuals for the TVET Certificate V in Networking and Internet Technologies, specifically for the module **"GENPP501: Python Programming Fundamentals".**

We extend our gratitude to KOICA Rwanda for its contribution to the development of these training manuals and for its ongoing support of the TVET system in Rwanda

We extend our gratitude to the TQUM Project for its financial and technical support in the development of these training manuals.

We would also like to acknowledge the valuable contributions of all TVET trainers and industry practitioners in the development of this training manual.

The management of Rwanda TVET Board extends its appreciation to both its staff and the staff of the TQUM Project for their efforts in coordinating these activities.

This training manual was developed:

Under Rwanda TVET Board (RTB) guiding policies and directives

Under Financial and Technical support of

COORDINATION TEAM

RWAMASIRABO Aimable

MARIA Bernadette M. Ramos

MUTIJIMA Asher Emmanuel

PRODUCTION TEAM

Authoring and Review

NGARAMBE François

NSABIMANA Jean Pierre

Validation

MUSABIREMA Janviere

MURWANASHYAKA Eric

Conception, Adaptation and Editorial works

HATEGEKIMANA Olivier

GANZA Jean Francois Regis

HARELIMANA Wilson

NZABIRINDA Aimable

DUKUZIMANA Therese

NIYONKURU Sylvestre

NGENDAHAYO Henry Gabriel

Formatting, Graphics, Illustrations, and infographics

YEONWOO Choe

SUA Lim

SAEM Lee

SOYEON Kim

WONYEONG Jeong

NDAYISABA Olivier

Financial and Technical support

KOICA through TQUM Project

TABLE OF CONTENT

ACRONYMS

API: Application Programming Interface

AWS: Amazon Web Services

CHMOD: Change Mode

CI/CD: Continuous Integration/Continuous Deployment

CSV: Comma-Separated Values

DB: Database

GUI: Graphical User Interface

I/O: Input/Output

IANA: Internet Assigned Numbers Authority

ID: Identification

IDE: Integrated Development Environment

IP: Internet Protocol

KOICA: Korea International Cooperation Agency

OOP: Object-Oriented Programming

OS: Operating System

PIP: Package Installer for Python

RAM: Random Access Memory

RTB: Rwanda TVET Board

SDK: Software Development Kit

SQL: Structured Query Language

SSH: Secure Shell

TQUM Project: TVET Quality Management Project

UTC: Coordinated Universal Time

VM: Virtual Machine

YAML: YAML Ain't Markup Language

INTRODUCTION

This trainer's manual includes all the methodologies required to effectively deliver the module title "**Python Programming Fundamentals."** Trainees enrolled in this module will engage in practical activities designed to develop and enhance their competencies.

The development of this training manual followed the Competency-Based Training and Assessment (CBT/A) approach, offering ample practical opportunities that mirror real-life situations.

The trainer's manual is organized into Learning Outcomes, which is broken down into indicative content that includes both theoretical and practical activities. It provides detailed information on the key competencies required for each learning outcome, along with the objectives to be achieved.

As a trainer, you will begin by asking questions related to the activities to encourage critical thinking and guide trainees toward real-world applications in the labor market. The manual also outlines essential information such as learning hours, didactic materials, and suggested methodologies.

This manual outlines the procedures and methodologies for guiding trainees through various activities as detailed in their respective trainee manuals. The activities included in this training manual are designed to offer students opportunities for both individual and group work. Upon completing all activities, you will assist trainees in conducting a formative assessment known as the end learning outcome assessment. Ensure that students review the key reading and the points to remember section.

MODULE CODE AND TITLE: GENPP501 PYTHON PROGRAMMING FUNDAMENTALS

Learning Outcome 1: Prepare Python Environment **Learning Outcome 2: Write basic Python Program** **Learning Outcome 3: Apply Object-Driven in Python**

Learning Outcome 1: Prepare Python Environment

```
Microsoft Windows [Version 10.0.22621.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\samsung>pip install nltk
Collecting nltk
    Downloading nltk-3.8.1-py3-none-any.whl (1.5 MB)

Collecting click
    Downloading click-8.1.7-py3-none-any.whl (97 kB)

Collecting joblib
    Downloading joblib-1.4.2-py3-none-any.whl (301 kB)

Collecting nltk
    Downloading nltk-3.8.1-py3-none-any.whl (97 kB)

Collecting regex>=2021.8.3
    Downloading regex-2024.10.22-cp311-win_amd64.whl(268 kB)

Collecting colorama
    Downloading colorama-0.4.6-py2-none-any.whl (25 kB)
Installing collected packages: regex, joblib, colorama, click, nltk
Successfully installed click-8.1.7 colorama-0.4.6 joblib-1.4.2 nltk-3.8.1 regex-2024.10.22

[notice] A new release of pip available : 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Indicative contents

1.1 Selection of Python tools

1.2 Installation of Python tools

1.3 Testing python installation

Key Competencies for Learning Outcome 1: Prepare Python Environment

Knowledge	Skills	Attitudes
● Description of python programming. ● Identifications of python tools. ● Identification of computer system requirements ● Description of application of python	● Installing python software tools ● Configuring python virtual environment ● Running python version command ● Checking python interpreter ● Testing package manager	● Having teamwork spirit ability ● Being critical thinker ● Being innovative ● Being attentive ● Being creative ● Problem solving ● Being practical oriented

Duration: 15 hrs

Learning outcome 1 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe correctly python programming as applied in software development.
2. Select properly python tools in accordance with computer operating system.
3. Identify properly computer system requirements in line with operating system.
4. Select correctly python tools depending on project to be developed.
5. Install correctly python software tools based on output of python version command.
6. Configure correctly python virtual environment based on operating system.
7. Test correctly python installation based on output of python version command.

Resources

Equipment	Tools	Materials
• Computer	• IDE (PyCharm) • Python interpreter (Latest version)	• Internet • Electricity

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have prepared computer lab.

Indicative content 1.1: Selection of Python tools

Duration: 5 hrs

Theoretical Activity 1.1.1: Description of python programming

Notes to the trainer:

- While delivering this content, small groups can be used for describing python programming.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

i. Define python
ii. Discuss on benefits of python
iii. Explain characteristics of python
iv. Python programming is applied in different areas, describe each of the following areas:
  a) Data science
  b) Software development
  c) Automation
  d) Data analysis

Step 2: Ask trainees to write their answers on papers or flipcharts
Step 3: Ask trainees to present their findings
Step 4: Provide expert view and clarification where necessary
Step 5: Address any questions or concerns
Step 6: Ask trainees to read the key reading 1.1.1 in the trainee manual

Points to Remember

- Python is a versatile programming language known for its simplicity and readability.
- Python has several benefits including: simplicity and readability, versatility, extensive libraries and frameworks, strong community support, cross-platform, compatibility, dynamic typing, interpreted language, integration capabilities, object-oriented and functional programming, rapid development.

- Key characteristics of python include: readability, versatility, efficiency and community and ecosystem.
- Python can be used in data science and machine learning by using the following frameworks: TensorFlow, PyTorch and Scikit-learn.
- Scientific Computing and Data Analysis Frameworks: NumPy, Pandas, Matplotlib.
- In python we can use the following automation Libraries: Selenium, PyAutoGUI, Requests, Beautiful Soup, Airflow, Celery, Paramiko, Fabric, Pywinauto, and Schedule.

Practical Activity 1.1.2: Selecting python tools

Notes to the trainer:

- This activity should take place in computer lab where trainees have to select python tools.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to perform the task described below: Ubumwe ltd need to develop a web application that help it in selling their products online and the system will have capabilities to perform automatic updates while products are sold and automatic deployment on the side of system administrator. You are hired as machine learning engineer responsible for selecting the best tools that will be used.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to select python tools, while demonstrating explain the criteria you based on while selecting python tools.

Step 4: Asks trainees to select python tools and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed and provide feedback if necessary.

Step 6: Ask trainees to read key reading 1.1.2.

Points to Remember

- Selecting the right Python tools for your project is crucial for productivity and quality. Start by defining your project's primary goal, whether it's web development, data analysis, or machine learning.
- Consider the tool's ease of use, integration with existing systems, scalability, and performance. A strong community and support, robust security features, and cost are also important factors.
- Ensure the tool is flexible, customizable, and compatible with your operating system and other tools you plan to use.
- Python support the following IDEs: VS Code, PyCharm, Jupyter Notebook, Spyder, Sublime Text, and Thonny
- Python support the following frameworks: Django, Flask, and FastAPI
- Python can be used in data science and machine learning by using the following frameworks: TensorFlow, PyTorch, and Scikit-learn
- Scientific Computing and Data Analysis Frameworks: NumPy, Pandas, and Matplotlib
- In python we can use the following automation Libraries: Selenium, PyAutoGUI, requests, BeautifulSoup, Airflow, Celery, Paramiko, Fabric, pywinauto, and Schedule.

Application of learning 1.1.

HH ltd want to develop a system that will be used while selling their products online and performing some automations once new features have added to their website and performing automatic deployment. You are hired as machine learning engineer responsible for selecting the right tools that will be used to develop that software.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python tools are well selected	IDE is selected		
	Python interpreter is selected		
	Automation library is selected		
	Web development framework is selected		
	Automation framework for deployment is selected		
Decision			

Indicative content 1.2: Installation of Python tools

Duration: 5 hrs

Theoretical Activity 1.2.1: Identification of computer system requirements

Notes to the trainer:

- While delivering this content, small groups can be used for identifying system requirements to install python tools.

Key steps:

While delivering this activity, pass through the following steps:

Step 1:	Introduce the activity and ask trainees to identify system to install python tools.
Step 2:	Ask trainees to write their findings on paper or flipchart
Step 3:	Ask trainees to present their findings
Step 4:	Provide expert view by using didactic materials
Step 5:	Address any questions or concerns
Step 6:	Ask trainees to read the key reading 1.2.1 in the trainee manual

Points to Remember

- To install Python tools, your system should meet certain hardware requirements: at least a dual-core processor, 4 GB RAM (preferably 8-16 GB for heavier tasks), 1 GB of free disk space, and optionally, a GPU for machine learning.
- On the software side, ensure you have the latest version of Python 3.x, an appropriate IDE (e.g., VS Code, PyCharm), pip for package management, and necessary dependencies like C++ build tools or Java for specific libraries

Practical Activity 1.2.2: Installation of python software tools

Notes to the trainer:

- This activity should take place in computer lab where trainees have to install python and PyCharm IDE.
- While delivering this content, you are required to:
  - ✓ Avail computer with windows OS installed.
  - ✓ Avail python and PyCharm IDE setup on external storage device.
  - ✓ Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As a machine learning engineer, you are asked to go to the computer lab to install python and PyCharm.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to install python and PyCharm check if python and PyCharm is installed.

Step 4: Asks trainees to perform the activity of step 4 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 1.2.2.

Points to Remember

- While installing python we follow these steps:

  Step 1: Download the Python Installer or use offline from external storage
  Step 2: Run the installer
  Step 3: Verify the installation

- While installing PyCharm we follow these steps:

  Step 1: Download PyCharm or use offline from external storage
  Step 2: Run the installer
  Step 3: Launch PyCharm

Practical Activity 1.2.3: Configuring python virtual environment

Notes to the trainer:

- This activity should take place in computer lab where trainees have to configure python virtual environment.
- While delivering this content, you are required to:
  - ✓ Avail computer with windows OS installed.
  - ✓ Avail python and PyCharm IDE installed on all computers to be used.
  - ✓ Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab to configure python virtual environment.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to configure python virtual environment, while demonstrating explain the steps to be followed.

Step 4: Asks trainees to also configure python virtual environment and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed. And provide feedback if necessary.

Step 6: Ask trainees to read key reading 1.2.3.

Points to Remember

- Python virtual environments create isolated environments for projects, preventing dependency conflicts.
- To create a virtual environment, use python -m venv myenv.
- Activate it using myenv\Scripts\activate.
- Install packages with pip install package_name.
- Deactivate using deactivate.

- Freeze dependencies with pip freeze > requirements.txt.
- Install from requirements.txt using pip install -r requirements.txt.

Application of learning 1.2.

HHT LTD is software Development Company located in Kicukiro district, they want to develop a web app that will help in selling their products online and the system will have capabilities to perform automatic deployment on the side of system administrator. You have been hired as machine learning engineer, responsible for installing and configuring all python tools that will be needed to develop that project.

The company will provide all tools, materials and equipment.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python software tools are well installed	Python is installed		
	PyCharm is installed		
	Installation is verified		
Python virtual environment is clearly configured	Virtual environment is created		
	Virtual environment is activated		
	Packages are installed in virtual environment		
	Is able to freeze all packages in requirements file.		
	Is able to install packages from requirements file.		
	Is able to deactivate virtual environment.		
Decision			

Indicative content 1.3: Testing python installation

Duration: 5 hrs

Practical Activity 1.3.1: Testing python installation

Notes to the trainer:

- This activity should take place in the computer lab where trainees should Run python version command, Check python interpreter and Test package manager.
- While delivering this content, you are required to:
  - ✓ Avail computers with python installed.
  - ✓ Avail videos to be used as didactic materials.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As machine learning engineer, you are asked to go to the computer lab to Run python version command, Check python interpreter and Test package manager.

Step 2: Explain the task and provide clear work instruction

Step 3: Demonstrate how to Run python version command, Check python interpreter and Test package manager. While demonstrating, explain the steps to be followed.

Step 4: Asks learners to Run python version command, check python interpreter, Test package manager, and monitor the procedures.

Step 5: Verify whether the tasks have been well performed and provide feedback if necessary

Step 6: Ask trainees to read key reading 1.3.1

Points to Remember

- To Check Python Version run that command in terminal: python --version or python3 --version
- To Check Python Interpreter run that command in terminal: python or python3
- To Test Package Manager: Check pip version pip –version and Install a package: pip install Django
- To list installed packages run: pip list

Application of learning 1.3.

HHT LTD is software Development Company located in Kicukiro district, they want to develop a web application that will help in selling their products online and the system will have capabilities to perform automatic deployment on the side of system administrator. You have been hired as machine learning engineer, responsible for testing the python installation and installing required packages that will be used.

The company will provide all tools, materials and equipment.

Checklist

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python installation is well tested	Python Version is checked		
	Python Interpreter is checked		
	pip version is checked		
	Package are installed		
	Installed packages are listed		
Decision			

Learning outcome 1 end assessment

Written assessment

I. Circle the letter corresponding to the right answer:

1. **Which of the following is a feature of Python?**
   a) Case sensitivity
   b) Curly-bracket syntax
   c) Indentation-based syntax
   d) Use of semicolons

ANSWER: c) Indentation-based syntax

2. **Python was created by:**
   a) Guido van Rossum
   b) James Gosling
   c) Bjarne Stroustrup
   d) Dennis Ritchie

ANSWER: a) Guido van Rossum

3. **Which Python framework is known for rapid development and scalability?**
   a) Flask
   b) Django
   c) FastAPI
   d) CherryPy

ANSWER: b) Django

4. **Python can be used for:**
   a) Web development
   b) Data analysis
   c) Machine learning
   d) All of the above

ANSWER: d) All of the above

5. **Which IDE is specifically designed for scientific computing?**
   a) PyCharm
   b) Visual Studio Code
   c) Spyder
   d) Sublime Text

ANSWER: c) Spyder

6. **Python is classified as a(n):**
   a) Compiled language
   b) Interpreted language
   c) Assembly language
   d) Machine language

ANSWER: b) Interpreted language

7. **What is the default file extension for Python scripts?**
   a) .pyc
   b) .txt
   c) .py
   d) .exe

ANSWER: c) .py

8. **The command to install a Python package using pip is:**
   a) python install package_name
   b) pip install package_name
   c) install package_name
   d) pip setup package_name

ANSWER: b) pip install package_name

9. **Which of the following is NOT a Python web framework?**
   a) Django
   b) Flask
   c) NumPy
   d) FastAPI

ANSWER: c) NumPy

10. **In Python, indentation is used to:**
    a) Declare variables
    b) Define the scope of loops and functions
    c) Import libraries
    d) Comment on the code

ANSWER: b) Define the scope of loops and functions

11. **Which Python library is primarily used for numerical computing?**
    a) Pandas
    b) NumPy
    c) Matplotlib
    d) Scikit-learn

ANSWER: b) NumPy

12. **Python's memory management is handled by:**
    a) The developer
    b) The Python interpreter automatically
    c) An external tool
    d) Manual memory allocation

ANSWER: b) The Python interpreter automatically

13. **Which of the following is an example of a Python text editor?**
    a) PyCharm
    b) Jupyter Notebook
    c) Sublime Text
    d) All of the above

ANSWER: d) All of the above

14. **Python was first released in:**
    a) 1989
    b) 1991
    c) 1995
    d) 2000

ANSWER: b) 1991

15. **What is the primary use of the TensorFlow library?**
    a) Web development
    b) Machine learning
    c) Data visualization
    d) Game development

ANSWER: b) Machine learning

16. **Which of the following is NOT a characteristic of Python?**
    a) Dynamic typing
    b) Complex syntax
    c) Object-oriented programming
    d) Readability

ANSWER: b) Complex syntax

17. **What does the command python --version do?**
    a) Runs a Python script
    b) Displays the current Python version installed
    c) Updates Python to the latest version
    d) Installs Python on your system

ANSWER: b) Displays the current Python version installed

18. **Which Python library is widely used for data visualization?**
    a) Matplotlib
    b) TensorFlow
    c) Flask
    d) Pandas

ANSWER: a) Matplotlib

19. **What does IDE stand for in the context of Python?**
    a) Integrated Development Environment
    b) Interactive Development Editor
    c) Integrated Debugging Environment
    d) Interactive Design Editor

ANSWER: a) Integrated Development Environment

20. **Which Python command is used to create a virtual environment?**
    a) python create venv
    b) python -m venv
    c) venv create
    d) create venv python

ANSWER: b) python -m venv

21. **Which Python framework is designed for asynchronous networking?**
    a) Kivy
    b) Twisted
    c) CherryPy
    d) Flask

ANSWER: b) Twisted

22. **In Python, a function is defined using the keyword:**
    a) func
    b) function
    c) def
    d) define

ANSWER: c) def

23. **Which IDE is web-based and primarily used for data analysis?**
    a) PyCharm
    b) Jupyter Notebook
    c) Visual Studio Code
    d) Thonny

ANSWER: b) Jupyter Notebook

24. **The command to deactivate a Python virtual environment is:**
    a) end venv
    b) stop venv
    c) deactivate
    d) exit venv

ANSWER: c) deactivate

25. **What type of language is Python?**
    a) Low-level
    b) High-level
    c) Mid-level

d) Machine-level

ANSWER: b) High-level

II. **Complete the following statements by using one of the keyword listed below: You can use one keyword once or more**

Display installed packages, Kivy, Large, Data type, IDEs, Delete, Functional, TensorFlow, Include libraries, Interpreted, pip, Indentation, FastAPI, Pandas, Beginners

1. Python is an ____________ language, meaning it executes code line by line.
2. The Python package manager is called ____________.
3. Python relies on ____________ to define the scope of loops, functions, and classes.
4. ____________ is a popular Python framework for building APIs, known for its performance and ease of use.
5. The Python library ____________ is used for data manipulation and analysis.
6. Python's simple and readable syntax makes it especially beneficial for ____________.
7. The command pip list is used to ____________.
8. ____________ is a Python framework used for developing cross-platform mobile and desktop applications.
9. Python's ____________ community provides extensive support, documentation, and resources.
10. In Python, variables do not require an explicit ____________ declaration.
11. PyCharm and VS Code are examples of ____________ used for Python development.
12. The command rmdir /s myenv is used to ____________ a Python virtual environment.
13. Python supports both object-oriented and ____________ programming paradigms.
14. The Python library ____________ is widely used for machine learning and deep learning.
15. In Python, the import statement is used to ____________.

ANSWER:

1. Interpreted
2. pip
3. Indentation
4. FastAPI
5. Pandas
6. Beginners
7. Display installed packages
8. Kivy
9. Large
10. Data type
11. IDEs
12. Delete
13. Functional

14. TensorFlow
15. Include libraries

III. Matching questions

1. Match the IDE in column A with their corresponding primary features in column B

Answers	**Column A**	**Column B**
a)……..	a) PyCharm	(i) Beginner friendly interface
b)……..	b) Jupyter Notebook	(ii) Intelligent code completion
c)……..	c) Thonny	(iii) Interactive environment for data analysis
d)……..	d) Visual Studio Code	(iv) Lightweight and customizable

ANSWER:
a) PyCharm –(i)
b) Jupyter Notebook –(ii)
c) Thonny –(iii)
d)Visual Studio Code–(iv)
16. **Match the Python framework in column A with its corresponding description in column B**

Answers	**Column A**	**Column B**
a)……..	a) Django	(i) Framework for cross platform applications
b)…….	b) Flask	(ii) Lightweight framework for small projects
c)…….	c) FastAPI	(iii) Full stack framework for web development
d)……	d) Kivy	(iv) High performance API framework

ANSWER:
a) Django –(i)
b) Flask –(ii)
c) FastAPI–(iii)
d) Kivy–(iv)

2. **Match the Python library of column A with its corresponding application in column B**

Answers	**Column A**	**Column B**
a)……..	a) NumPy	(i) Numerical computing
b)……..	b) Pandas	(ii) Data manipulation
c)……..	c) Matplotlib	(iii) Data visualization
d)……..	d) Scikit learn	(iv) Machine learning

ANSWER:

a) NumPy –(i)

b) Pandas –(ii)

c) Matplotlib –(iii)

d) Scikit learn –(iv)

3. **Match the Python characteristic in column A with its corresponding feature of column B**

Answers	**Column A**	**Column B**
a)……..	a) Dynamic typing	(i) Clear syntax and structure
b)……..	b) Readability	(ii) No explicit data type declaration
c)……..	c) Efficiency	(iii) Large standard library
d)……..	d) Versatility	(iv) Wide range of applications

ANSWER:

a) Dynamic typing –(i)

b) Readability –(ii)

c) Efficiency –(iii)

d) Versatility–(iv)

4. **Match the Python command in column A with its corresponding function of column B**

Answers	**Column A**	**Column B**
a)……..	a) python --version	(i) Display the Python version
b)……..	b) pip install	(iii) Install Python packages
c)……..	c) deactivate	(iv) Exit the virtual environment
d)……..	d) pip list	(ii) Display installed packages

ANSWER:

a) python --version –(i)

b) pip install –(ii)

c) deactivate –(iii)

d) pip list–(iv)

5. **Match the Python tools of column A with its corresponding description in column B**

Answers	Column A	Column B
a)……..	a) PyTorch	(i) Computer vision library
b)……..	b) TensorFlow	(ii) Flexible machine learning framework
c)……..	c) SciPy	(iii) Platform for deep learning
d)……..	d) OpenCV	(iv) Scientific computing and statistics

ANSWER:

a) PyTorch –(i)

b) TensorFlow –(ii)

c) SciPy –(iii)

d) OpenCV–(iv)

6. **Match the Python task with the corresponding suitable library/framework:**

Answers	Python Task	Library
a)……..	a) Web Development	(i) Scikit learn
b)……..	b) Data Visualization	(ii) Matplotlib
c)……..	c) Machine Learning	(iii) Django
d)……..	d) Task Automation	(iv) Python’s standard library

ANSWER:

a) Web Development –(i)

b) Data Visualization –(ii)

c) Machine Learning –(iii)

d) Task Automation–(iv)

7. **Match the Python version command in Column A with its corresponding output in Column B:**

Answers	Column A	Column B
a)……..	a) python --version	(i) Displays the Python version number
b)……..	b) pip show numpy	(ii) Shows details of the installed NumPy package
c)……..	c) python -m venv env	(iii) Creates a new virtual environment
d)……..	d) pip freeze	(iv) Lists installed packages in the virtual environment

ANSWER:

a) python --version -(i)

b) pip show numpy -(ii)

c) python -m venv env -(iii)

d) pip freeze -(iv)

8. **Match the Python command in Column A with the corresponding action it performs in Column B:**

Answers	**Column A**	**Column B**
a)........	a) import	(i) Defines a function
b)........	b) def	(ii) Imports a module or library
c)........	c) print	(iv) Outputs data to the console
d)........	d) class	(iii) Defines a new class

ANSWER:
a) import –(i)
b) def –(ii)
c) print –(iii)
d) class –(iv)

Match the following Python versions in Column A with their corresponding key characteristics in Column B:

Answers	**Column A**	**Column B**
a)........	a) Python 2.x	(i) Legacy version with different print syntax
b)........	b) Python 3.x	(ii) Current version with updated syntax
c)........	c) Python 3.6+	(iv) Introduced f string formatting
d)........	d) Python 3.8+	(iii) Introduced assignment expressions

ANSWER:
a) Python 2.x –(i)
b) Python 3.x –(ii)
c) Python 3.6+ –(iii)
d) Python 3.8+ –(iv)

Practical assessment

HHT LTD is software Development Company located in Kicukiro district, they want to develop a web application that will help in selling their products online and the system will have capabilities to perform automatic deployment on the side of system administrator and automatic updates once new feature is added. You have been hired as machine learning engineer, responsible for installing, configuring all python tools that will be needed to develop that project, testing the python installation and installing required packages that will be used.

The company will provide all tools, materials and equipment.

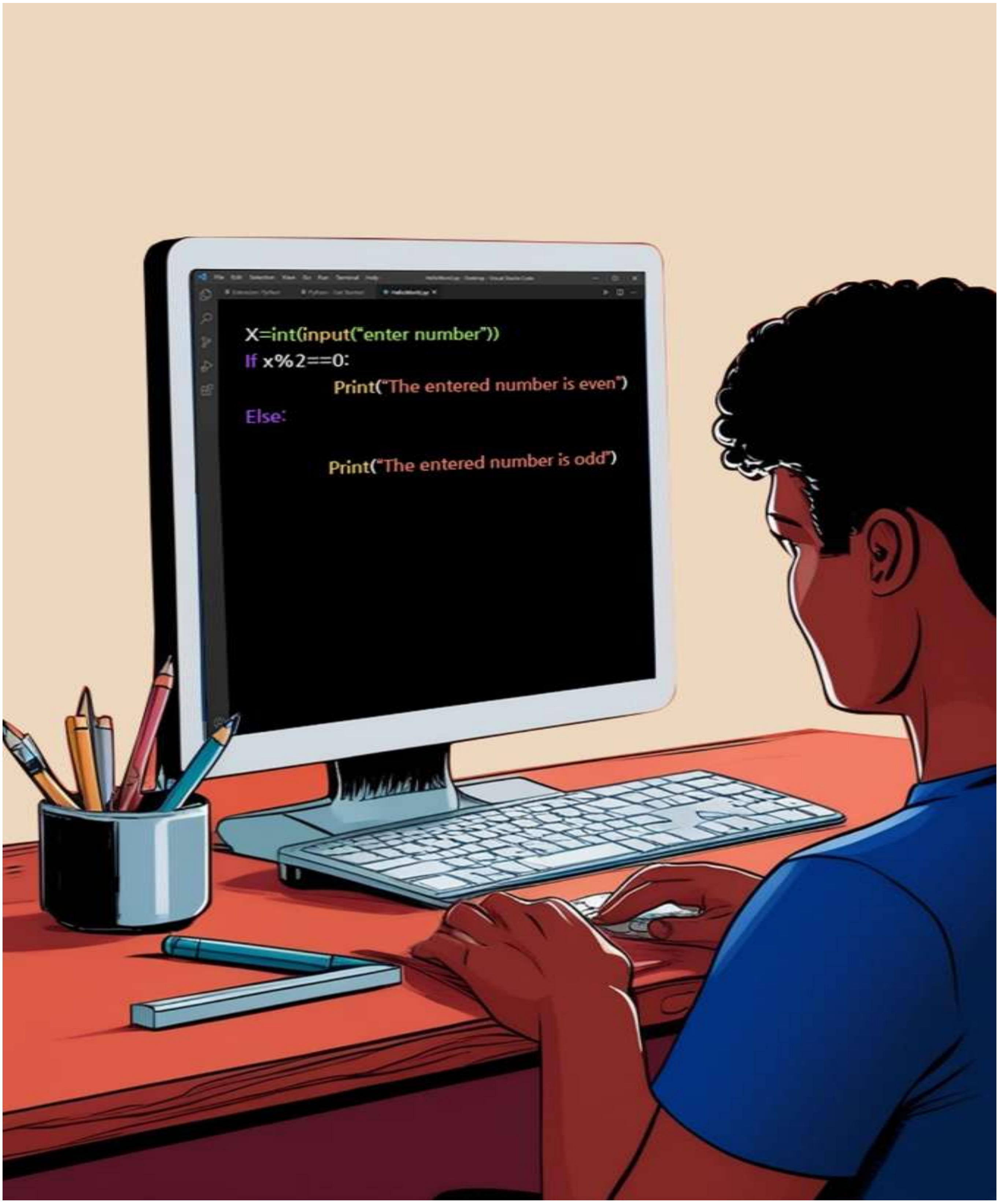
Checklist:

Criteria	Indicators	Observation	
		Yes	No
Python software tools are well installed	Python is installed		
	PyCharm is installed		
	Installation is verified		
Python virtual environment is clearly configured	Virtual environment is created		
	Virtual environment is activated		
	Packages are installed in virtual environment		
	Is able to freeze all packages in requirements file		
	Is able to install packages from requirements file		
	Is able to disactivate virtual environment		
Python installation is well tested	Python Version is checked		
	Python Interpreter is checked		
	pip version is checked		
	Package are installed		
	Installed packages are listed		
Decision			

Further information to the trainer

JetBrains. (n.d.). Configure a virtual environment. Retrieved from https://www.jetbrains.com/help/pycharm/configuring-python-interpreter.html visited on 19.10.2024
JetBrains. (n.d.). Configure a Python interpreter. Retrieved from https://www.jetbrains.com/help/pycharm/configuring-python-interpreter.html visited on 19.10.2024
JetBrains. (n.d.). Quick start guide. Retrieved from https://www.jetbrains.com/help/pycharm/quick-start-guide.html visited on 19.10.2024
Python Software Foundation. (n.d.). Installing Python. Retrieved from https://www.python.org/downloads/ visited on 19.10.2024
JetBrains. (n.d.). Install PyCharm. Retrieved from https://www.jetbrains.com/pycharm/download/ visited on 19.10.2024
https://www.youtube.com/watch?v=kqtD5dpn9C8
https://www.coursera.org/specializations/python
https://cs50.harvard.edu/python/2022/
https://www.coursera.org/specializations/python
https://docs.python.org/3/
https://automatetheboringstuff.com/

Learning Outcome 2: Write Basic Python Program



```
X=int(input(“enter number”))
If x%2==0:
            Print(“The entered number is even”)
Else:

            Print(“The entered number is odd”)
```

Indicative contents

2.1 Applying python basic concepts
2.2 Applying python control structures
2.3 Applying functions in Python
2.4 Applying of Python Collections
2.5 Performing File handling

Key Competencies for Learning Outcome 2: Write Basic Python Program

Knowledge	Skills	Attitudes
● Description of python basic concepts ● Description of function in python ● Description of python Collections ● Description of file Handling libraries	● Applying python basic concepts ● Applying conditional Statements ● Applying looping Statements ● Using Jump Statements ● Creating function in python ● Applying special purpose functions ● Applying Python Collections ● Performing operations on collection ● Practicing read file ● Performing write/create and delete file	● Having teamwork spirit ability while coding ● Being critical thinker in logic of coding ● Being innovative in coding ● Being attentive ● Being creative in discovering new logics ● Having a problem solving ● Being practical oriented

	Applying python best practices	

Duration: 45 hrs

Learning outcome 2 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe correctly python basic concepts based on python standards
2. Describe properly function in python based on python standards
3. Describe correctly python Collections based on python standards
4. Describe properly file Handling libraries based on python standards
5. Apply correctly python basic concepts based on python standards
6. Apply correctly conditional Statements based on python standards
7. Apply correctly looping Statements based on python standards
8. Use correctly Jump Statements based on python standards
9. Create properly function in python based on python standards
10. Apply properly special purpose functions based on python standards
11. Apply correctly Python Collections based on python standards
12. Perform correctly operations on collection based on python standards
13. Practice clearly read file based on python standards
14. Perform correctly write/create and delete file based on python standard
15. Apply correctly python best practices based on python standards

Resources

Equipment	**Tools**	**Materials**
● Computer	● Python ● Python IDE (Pycharm)	● Internet

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have computers installed with python and PyCharm or any other IDE that support python.

Indicative content 2.1: Applying python basic concepts

Duration: 9 hrs

Theoretical Activity 2.1.1: Description of python basic concepts

Notes to the trainer:

- While delivering this activity small groups can be used to discuss python basic concepts.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to describe the following python basic concepts:

i. Data types
ii. Variable
iii. Comments
iv. Operators

Step 2: Ask trainees to write their answers on paper or flipchart

Step 3: Ask trainees to present their findings

Step 4: Provide expert view by using didactic materials

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 2.1.1 in the trainee manual

Points to Remember

- Variables are containers that store data values.
- Data types define the kind of values a variable can hold.
- Python supports numeric, sequence, mapping, set, and Boolean data types.
- Comments are used to explain code.
- Operators are symbols used to perform operations on variables and values.

Practical Activity 2.1.2: Applying python basic concepts

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply python basic concepts.
- While delivering this content the trainer have to:
  - ✓ Avail computer installed with python and PyCharm.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab to apply python data types, variable, python comments and operators while developing a python program that can perform simple calculation add, subtract, multiply, divide and calculate the module and exponent for two entered numbers by user.
Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to apply python basic concepts while developing a simple calculator, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.1.2.

Points to Remember

- Data types define the kind of values a variable can hold.
- Python supports numeric, sequence, mapping, set, and Boolean data types.
- Variables are containers that store data values.
- Comments are used to explain code.
- Operators are symbols used to perform operations on variables and values.

Application of learning 2.1.

As machine learning engineer, you are asked to develop a python program that can perform simple calculation add, subtract, multiply, divide and calculate the module and exponent for two entered numbers by user. The developed program has to contain comments for better explanation.

Checklist:

Criteria	Indicators	Observation	
		Yes	No
Python basic concepts are well applied	Variables are declared		
	Datatypes are used		
	Comments are used		
	+ operator is used for addition		
	* operator is used for multiplication		
	/ operator is used for division		
	% is used for modulus		
	- Operator is used for subtraction		
Decision			

Solution/output:

```
# Simple Calculator Program
# This program performs basic arithmetic operations: addition, subtraction,
# multiplication, division, modulus, and exponentiation on two numbers.

def add(x, y):
"""Return the sum of x and y."""
return x + y


def subtract(x, y):
"""Return the difference of x and y."""
return x - y


def multiply(x, y):
"""Return the product of x and y."""
return x * y


def divide(x, y):
```

```
"""Return the quotient of x and y. Handles division by zero."""
if y == 0:
return "Error: Division by zero!"
return x / y


def modulus(x, y):
"""Return the modulus of x and y."""
return x % y


def exponent(x, y):
"""Return x raised to the power of y."""
return x ** y


# Main function to execute the calculator
def main():
   print("Welcome to the Simple Calculator!")

# Input: Getting two numbers from the user
try:
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
except ValueError:
print("Invalid input! Please enter numeric values.")
return

# Display the available operations
print("\nChoose an operation:")
print("1. Add")
print("2. Subtract")
print("3. Multiply")
print("4. Divide")
print("5. Modulus")
print("6. Exponent")

# Input: Getting the user's choice
operation = input("Enter the operation (1/2/3/4/5/6): ")

# Performing the chosen operation and displaying the result
if operation == '1':
print(”{num1} + {num2} = {add(num1, num2)}")
elif operation == '2':
print("{num1} - {num2} = {subtract(num1, num2)}")
elif operation == '3':
print("{num1} * {num2} = {multiply(num1, num2)}")
```

```
elif operation == '4':
print(f"{num1} / {num2} = {divide(num1, num2)}")
elif operation == '5':
print(f"{num1} % {num2} = {modulus(num1, num2)}")
elif operation == '6':
print(f"{num1} ** {num2} = {exponent(num1, num2)}")
else:
print("Invalid operation! Please enter a number between 1 and 6.")


# Entry point of the program
if __name__ == "__main__":
main()
```

Indicative content 2.2: Applying python control structures

Duration: 9 hrs

Practical Activity 2.2.1: Applying Conditional Statements

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply conditional statement in python programs.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Avail video to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As machine learning engineer, you are asked to go to the computer lab to apply if, elif and else statement while developing a python program that can be used when displaying the grade of students depending on entered marks as requested by NESA once analysing marks and student placement.

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to apply if, elif and else, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step.7: Ask trainees to read key reading 2.2.1.

Points to Remember

- Conditional statements are used to execute different code blocks based on specific conditions.
- The "if statement" is used to execute a block of code if a condition is true.
- The "else statement" is used to execute a block of code if the if condition is false.
- The "elif statement" is used to test additional conditions if the previous if or elif conditions are false.
- Indentation is crucial for defining the code blocks within conditional statements.
- Comparison operators (e.g., ==, !=, <, >, <=, >=) are used to create conditions.
- Logical operators (e.g., and, or, not) can be used to combine multiple conditions.
- Nested conditional statements can be used to create more complex decision-making logic.
- Proper indentation is essential for ensuring correct code execution.
- Testing different conditions can help verify the correctness of conditional statements.

Practical Activity 2.2.2: Applying Looping Statements

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply looping statement in python programs.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Avail video to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab to apply looping statement while writing python programs.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to apply loping statement, while demonstrating explain the code to be used.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 2.2.2.

Points to Remember

- **Looping statements** are used to repeat a block of code multiple times in Python.
- The “for loop” is used to iterate over a sequence of elements (e.g., lists, tuples, strings).
- The “while loop” is used to repeat a block of code as long as a condition is true.
- The **break statement** can be used to exit a loop prematurely.
- The **continue statement** can be used to skip the current iteration of a loop and move to the next one.

- **Nested loops** can be used to create more complex looping structures.
- **Indentation** is crucial for defining the code block within loops.

Practical Activity 2.2.3: Using Jump Statements

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to use jump statements.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Avail video to be used as didactic material

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab to apply looping statement while writing python programs and inside looping statement use break, continue and pass statements.

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to use jump statement in loop, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.2.3.

Points to Remember

- **“break”:** Terminates the loop entirely when a condition is met.
- **“continue”**: Skips the current iteration and continues with the next one.
- **“pass”:** Does nothing and is useful for maintaining the structure of code where a statement is syntactically required.

Application of learning 2.2.

Write a python program that can display all even numbers from 0 to 100 and the program skip 60 and 80 and stops execution if value is equal to 90.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python control structures are well applied	Loop is used		
	Continue is used		
	Break is used		
	Conditions are well applied		
Decision			

Solution/Output:

```
for number in range(0, 101):
if number == 90:  break
if number % 2 == 0 and number not in (60, 80):
print(number)
```

Explanation:

- The program uses a for loop to iterate through numbers from 0 to 100.
- It checks if a number is even using the modulus operator (%).
- It skips the numbers 60 and 80 by checking if the number is not in the tuple (60, 80).
- If the conditions are met, the number is printed.

Indicative content 2.3: Applying functions in Python

Duration: 9 hrs

Theoretical Activity 2.3.1: Description of function in python

Notes to the trainer:

While delivering this activity small groups can be used to describe functions in python programming.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to answer the followings questions related to functions in python programming:

i. Define function in python
ii. Differentiate two (2) types of function in python
iii. Elaborate characteristics and Advantages of Functions in python

Step 2: Ask trainees to write their answers on paper or flipchart
Step 3: Ask trainees to present their findings
Step 4: Provide expert view by using didactic materials
Step 5: Address any questions or concerns
Step 6: Ask trainees to read the key reading 2.3.1 in the trainee manual

Points to Remember

- Functions are fundamental building blocks in Python programming. They promote code reuse, readability, and maintainability.
- Built-in Functions: These are functions that are pre-defined in Python and can be used without any additional code.
- User-Defined Functions: These are functions that you define yourself to perform specific tasks. You can create them using the `def` keyword.

- Characteristics of Functions include: Modularity, Reusability, Parameters and Return Values, Encapsulation and Scope.
- Advantages of Functions include: Improved Readability, Easier Maintenance, Code Reusability, Debugging and Abstraction.

Practical Activity 2.3.2: Creating function in python

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to create functions in python.
- While delivering this content the trainer has to:
    - ✓ Avail computer installed with python and PyCharm.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are requested to go to the computer lab to create a function with arguments, default parameter value, passing a list as an argument and calling a function.

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to create functions in python, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.3.2.

Points to Remember

Creation of Function in Python you can follow the following steps:
Step 1: Defining a Function
Step 2: Arguments
Step 3: Default Parameter Value

Step 4: Passing a List as an Argument
Step 5: Calling a Function

Practical Activity 2.3.3: Applying special purpose functions

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply special purpose functions.
- While delivering this content the trainer has to:

✓ Avail computer installed with python and PyCharm.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are requested to go to the computer lab to apply special purpose functions such as Lambda, Python Generators, Python Closures, Python Decorators, Recursive function, and Higher-order function.
Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to apply special purpose functions, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.3.3.

Points to Remember

- Special purpose functions offer unique capabilities and can be used to solve specific problems.
- Lambda functions are concise and often used for short, simple expressions.
- Generators provide efficient ways to generate values on demand.

- Closures can be used to create functions with state.
- Decorators can modify the behavior of other functions without directly changing their code.
- Recursive functions can be used to solve problems that can be broken down into smaller, similar sub problems.
- Higher-order functions can be used to create more flexible and reusable code.

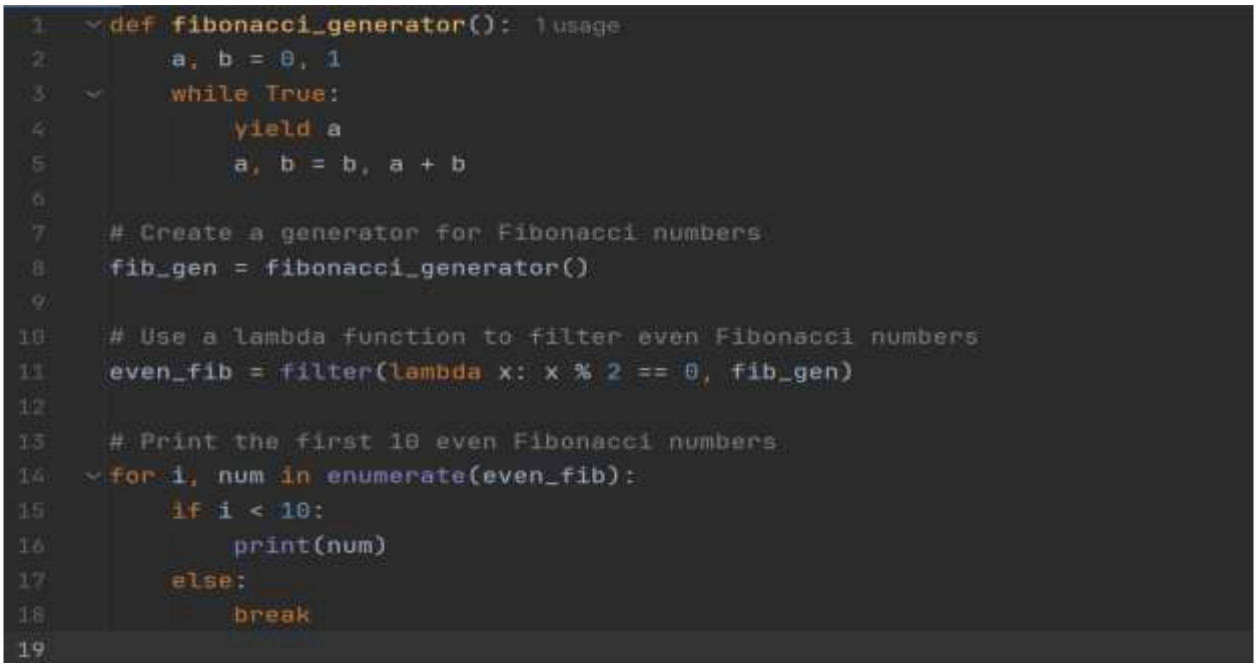
Application of learning 2.3.

Write a Python program that defines a function that generates Fibonacci numbers using a generator, uses a lambda function to filter even numbers from the generated sequence then prints the first 10 even Fibonacci numbers.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python functions are well applied	The Fibonacci generator function is defined		
	A lambda function is used to filter even Fibonacci numbers.		
	The filtering logic correctly identifies even numbers.		
	The loop correctly iterates over the filtered results and prints the first 10.		
	The output contains the correct first 10 even Fibonacci numbers.		
Decision			

Solution/Output:



```python
def fibonacci_generator():  1 usage
    a, b = 0, 1
    while True:
        yield a
        a, b = b, a + b

# Create a generator for Fibonacci numbers
fib_gen = fibonacci_generator()

# Use a lambda function to filter even Fibonacci numbers
even_fib = filter(lambda x: x % 2 == 0, fib_gen)

# Print the first 10 even Fibonacci numbers
for i, num in enumerate(even_fib):
    if i < 10:
        print(num)
    else:
        break

```

Explanation:

- Fibonacci Generator: The fibonacci_generator function yields Fibonacci numbers indefinitely.
- Filtering Even Numbers: A list comprehension generates the first 100 Fibonacci numbers, and then filter with a lambda function is used to select only the even numbers.
- Output: Finally, it prints the first 10 even Fibonacci numbers.

Indicative content 2.4: Applying Python Collections

Duration: 9hrs

Theoretical Activity 2.4.1: Description of python Collections

Notes to the trainer:

- While delivering this activity small groups can be used to describe python collections.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to describe the concepts of OOP below:
  i. Describe collection Types in Python
  ii. Explain Tools from the Collections Module

Step 2: Ask trainees to write their answers on paper or flipchart

Step 3: Ask trainees to present their findings

Step 4: Provide expert view by using didactic materials

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 2.4.1 in the trainee manual

Points to Remember

- Python provides various built-in collection types, each serving different purposes: Lists, tuples, dictionaries, sets, frozen sets, ChainMaps, and deques.
- The collections module enhances functionality with specialized tools like: Counter, OrderedDict, and defaultdict.

Practical Activity 2.4.2: Performing common operations on collection.

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to perform operations on collections.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Avail sample videos to be used as didactic materials.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer , you are requested to go to the computer lab to perform the following operations on collections:

i. Adding and Removing Elements
ii. Accessing and Iterating Over Elements
iii. Filtering and Sorting
iv. Set Operations and Counting
v. Stack and Queue Operations

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to create functions in python, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.3.2.

Points to Remember

- These are common operations you can perform on various collection types in Python. Lists, dictionaries, sets: Adding and removing elements, accessing and iterating over elements, filtering and sorting, set operations and counting, stack and queue operations.
- Specialized tools such as deques provide powerful ways to manage and manipulate data, making Python a versatile language for handling collections.

Application of learning 2.4.

Write a Python program that creates a list of numbers, filters out even numbers, sorts the remaining numbers in ascending order and then prints the sorted list.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python collections are well applied	The program creates a list of numbers		
	The filtering logic identifies and retains only odd numbers (i.e. x % 2 != 0)		
	A lambda function is used effectively within the filter function		
	The program uses the sorted function to sort the list in ascending order		
	The program prints the sorted list of odd numbers		
Decision			

Solution/output:

```
#Create a list of numbers

numbers = [12, 7, 5, 10, 3, 14, 9, 8, 1, 6]

# Filter out even numbers using a list comprehension

odd_numbers = [num for num in numbers if num % 2 != 0]

# Sort the remaining numbers in ascending order

sorted_odd_numbers = sorted(odd_numbers)

# Print the sorted list

print(sorted_odd_numbers)
```

Explanation:

1. List Creation: A list of numbers is defined.
2. Filtering: A list comprehension filters out even numbers (keeping only odd numbers).
3. Sorting: The sorted() function is used to sort the filtered list in ascending order.
4. Output: The sorted list of odd numbers is printed.

Indicative content 2.5: Performing File handling

Duration: 9hrs

Theoretical Activity 2.5.1: Description of file handling libraries

Notes to the trainer:

- While delivering this activity small groups can be used to describe file handling libraries.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to describe the following file handling libraries:

i. os module
ii. pathlib module
iii. shutil module
iv. pandas library

Step 2: Ask trainees to write their answers on paper or flipchart
Step 3: Ask trainees to present their findings
Step 4: Provide expert view by using didactic materials
Step 5: Address any questions or concerns
Step 6: Ask trainees to read the key reading 2.5.1 in the trainee manual

Points to Remember

- Os library used for interacting with the operating system and file system;
- Pathlib library used for an object-oriented approach to path manipulation;
- Shutil library used for high-level file operations such as copying and moving files;
- Pandas library used for reading and writing data in various formats, primarily used for data analysis.

Practical Activity 2.5.2: Practicing open and read file

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to practice read file.
- While delivering this content the trainer has to:
    - ✓ Avail computer installed with python and PyCharm.
    - ✓ Avail a text file that will be used to implement those practice.
    - ✓ Avail sample videos to be used as didactic materials.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are requested to go to the computer lab to perform the following operations on file:

i. Open a File

ii. Read File Permissions

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to open and read file permission, while demonstrating explain the code to be used.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 2.5.2.

Points to Remember

- Opening a File: Use the open() function with the appropriate mode to read from or write to a file.
- Reading a File: You can read the entire content using file.read(), or read line by line using file.readline() or file.readlines().
- Checking Permissions: Use the os module to check whether you have read, write, or execute permissions for a file before attempting to open it.

Practical Activity 2.5.3: Performing write/create and delete file

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to write, create and delete file.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Avail sample videos to be used as didactic materials.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are requested to go to the computer lab to perform the following operations on file:

i. Write
ii. Create
iii. Delete file

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to write, Create and delete file, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.5.3.

Points to Remember

- Creating a New File: Use open(file_path, 'w') or open(file_path, 'x') to create a new file.
- Writing to an Existing File: Use open(file_path, 'a') to append or 'w' to overwrite.
- Removing a File: Use os.remove() to delete a file.

- Deleting a Folder: Use os.rmdir() for empty directories and shutil.rmtree() for non-empty directories.

Practical Activity 2.5.4: Applying python best practices

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply python best practices.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Avail sample project created in python to be used.
  - ✓ Avail sample videos to be used as didactic materials.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer developer, you are requested to go to the computer lab to apply the following:

i. Readability and Style
ii. Use of Built-in Features
iii. Efficiency and Memory Usage
iv. Error Handling and Testing

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to apply python best practices, while demonstrating explain the code to be used.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 2.5.4.

Points to Remember

- Adhering to Python best practices can improve code quality, readability, and maintainability.
- Following PEP 8 guidelines promotes consistent coding style.
- Using built-in features can make code more concise and efficient.
- Writing efficient code can reduce resource consumption.
- Proper error handling can prevent unexpected program failures.
- Testing can help identify and fix bugs early in the development process.
- Regular code reviews can help improve coding practices and catch potential issues.

Application of learning 2.5

Develop a python program that can create a file in excel format "list of TVET schools" and attach the following as header (District, School name, Trade, Number of students) and attach sample data in row 1 (Your district, Your school name, SWD,23). The created file has to be saved on desktop of your computer in directory works if that directory don't exist it has to create it before saving that file.

Checklist

SN	Criteria	Indicator	Observation	
			Yes	No
1	Python collections are well applied	1.1 The necessary package is installed		
		1.2 The file is created		
		1.3 The directory is created on Desktop using pathlib library		
		1.4 The os library have been used		
		1.5 The header contents are inserted		
		1.6 The sample data have been added		

		1.7 The program is proving the correct output		

Solution/Output:

```
import openpyxl

import os

from pathlib import Path

# Define the directory path where the file should be saved

desktop_path = Path.home() / 'Desktop' / 'Works'

# Create the directory if it doesn't exist

if not desktop_path.exists():

    desktop_path.mkdir(parents=True, exist_ok=True)

# Create a new workbook and select the active sheet

workbook = openpyxl.Workbook()

sheet = workbook.active

# Define the headers

headers = ["District", "School Name", "Trade", "Number of Students"]

# Add the headers to the first row

sheet.append(headers)

# Add sample data to the second row

sample_data = ["Nyanza", "nyanza TSS", "SWD", 234]

sheet.append(sample_data)

# Define the file path where the Excel file will be saved

file_name = desktop_path / "list_of_TVET_schools.xlsx"

# Save the workbook to the file

workbook.save(file_name)

# Print a success message

print(f"Excel file '{file_name.name}' created successfully on your Desktop in the 'Works' directory!")
```

```
# Display the contents of the created Excel file
print("\nContents of the file:")
workbook = openpyxl.load_workbook(file_name)
sheet = workbook.active
# Iterate through the rows and print each row
for row in sheet.iter_rows(values_only=True):
print(row)
```

Key Points:

- ✓ **Path Handling:** Path.home() is used to get the home directory of the current user, and Desktop is appended to it to form the path to the desktop.
- ✓ **Directory Creation:** desktop_path.mkdir(parents=True, exist_ok=True) ensures that the "Works" directory is created if it doesn't exist.
- ✓ **Saving the File:** The Excel file is saved in the "Works" directory on the desktop as list_of_TVET_schools.xlsx.
- ✓ **Reading and Displaying File Contents:** The contents of the Excel file are read and displayed in the terminal using openpyxl.

Expected Output:

When the program runs, it will:

- ✓ Create the "Works" directory on your desktop if it doesn't exist.
- ✓ Save the Excel file list_of_TVET_schools.xlsx with headers and sample data.

Display the contents of the created Excel file in the terminal

Learning outcome 2 end assessment

Written assessment

I. Circle the letter that corresponds to the right answer

1. What is the correct syntax to create a list in Python?

   A) list = (1, 2, 3)
   B) list = [1, 2, 3]
   C) list = {1, 2, 3}
   D) list = <1, 2, 3>

ANSWERS: B) list = [1, 2, 3]

2. Which of the following is an immutable data type in Python?

   A) List
   B) Dictionary
   C) Tuple
   D) Set

ANSWERS: C) Tuple

3. What will the output of the following code be?

```
x = 5
y = 10
print(x > y)
```

   A) True
   B) False
   C) 5
   D) 10

ANSWER: B) False

4. Which operator is used for exponentiation in Python?

   A) ^
   B) **
   C) //
   D) ***

ANSWER: B) **

5. Which keyword is used to define a function in Python?

    A) function
    B) def
    C) define
    D) func

ANSWER: B) def

6. What does the 'break' statement do in a loop?

    A) Skips the current iteration
    B) Exits the loop
    C) Repeats the loop
    D) Terminates the program

ANSWER: B) Exits the loop

7. How do you create a dictionary in Python?

    A) dict = [key: value]
    B) dict = {key: value}
    C) dict = (key: value)
    D) dict = <key: value>

ANSWER: B) dict = {key: value}

8. Which function is used to read a CSV file into a DataFrame using pandas?

    A) pd.read_csv()
    B) pd.load_csv()
    C) pd.import_csv()
    D) pd.open_csv()

ANSWER: A) pd.read_csv()

9. What will len([1, 2, 3]) return?

    A) 2
    B) 3
    C) 4
    D) None

ANSWER: B) 3

10. Which of the following is a built-in function in Python?

    A) print()
    B) show()
    C) display()

D) output()

ANSWER: A) print()

11. In Python, what data type is used to represent True or False values?

    A) int
    B) float
    C) bool
    D) str

ANSWER: C) bool

12. What will the output of the following code snippet?

```
print("Hello, World!"[7])
```

    A) H
    B) e
    C) W
    D) o

ANSWER: C) W

13. Which of the following is NOT a valid way to comment in Python?

    A) # This is a comment
    B) /* This is a comment */
    C) """ This is a comment """
    D) #!

ANSWER: B) /* This is a comment */

14. What does the 'continue' statement do in a loop?

    A) Exits the loop
    B) Skips to the next iteration
    C) Restarts the loop
    D) Ends the program

ANSWER: B) Skips to the next iteration

15. Which of the following is used to create a set in Python?

    A) []
    B) ()
    C) {}
    D) <>

ANSWER: C) {}

16. What is the correct way to define a lambda function in Python?

    A) lambda x, y: x + y
    B) function x, y: x + y
    C) def x, y: x + y
    D) x, y -> x + y

ANSWER : A) lambda x, y: x + y

17. What is the output of print(type(3.14))?

    A) <class 'int'>
    B) <class 'float'>
    C) <class 'str'>
    D) <class 'bool'>

ANSWER: B) <class 'float'>

18. Which of the following statements is used to import the os module?

    A) import os
    B) include os
    C) using os
    D) require os

ANSWER: A) import os

19. What will the following code output?

```
x = 20
if x < 10:
print("Small")
else:
print("Large")
```

    A) Small
    B) Large
    C) 20
    D) None

ANSWER: B) Large

20. Which method can be used to add an item to a list in Python?

    A) add()
    B) append()
    C) insert()
    D) Both B and C

ANSWER: D) Both B and C

1. **Complete the following statements by correct word, operator, or keword from the listed ones :**

(function, ==, loop, lambda, os, defaultdict, break, tuple, open, dictionary)

1. A ________ is a block of reusable code that performs a specific task in Python.
2. The __________ operator is used to compare two values for equality.
3. In Python, a __________ allows you to iterate over a sequence.
4. A __________ function can take any number of arguments but can only have one expression.
5. The __________ module provides a way to use operating system-dependent functionality in Python.
6. A __________ is a dictionary subclass that provides a default value for a non-existent key.
7. The __________ statement is used to exit the nearest enclosing loop in Python.
8. A __________ is an immutable collection that can hold a variety of object types.
9. The __________ function is used to read the contents of a text file.
10. A __________ is a collection of key-value pairs where keys must be unique.

ANSWER:

1. function
2. ==
3. loop
4. lambda
5. os
6. defaultdict
7. break
8. tuple
9. open
10. Dictionary

II. Answer by True or False to the followings:

1. A list in Python is immutable.
2. A function in Python can return multiple values.
3. The 'else' clause can be used with a 'for' loop.
4. The pandas library is primarily used for file handling.
5. Sets in Python can contain duplicate elements.
6. The 'pass' statement in Python does nothing when executed.
7. You can use the 'with' statement for file handling in Python.
8. A frozen set is a mutable version of a set.
9. The 'elif' keyword is used to check multiple conditions in Python.
10. Variables in Python do not require a declaration before use.

ANSWERS:

1. False
2. True
3. True
4. False
5. False
6. True
7. True
8. False
9. True
10. True

III. Match the following Python data types with their corresponding descriptions:

Answers	Data Type	Description
1………	1. List	A. Unordered collection of unique elements
2………	2. Tuple	B. Ordered, mutable collection
3………	3. Dictionary	C. Key-value pairs
4………	4. Set	D. Ordered, immutable collection

ANSWER:

1. List -A
2. Tuple – B
3. Dictionary – C
4. Set – D

Practical assessment

CODEX DEV LTD, a Kigali-based Software Development company, is seeking a machine learning engineer to implement a shopping cart feature in their existing system. The developer will be responsible for receiving a list of items to be purchased, allowing users to remove items from the cart, and providing an option to empty the cart entirely. And app must store that cart information to file named cart.csv The project will be implemented using Python.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python collection is well applied	Cart item is added		
	Items can be removed from cart		
	Cart items is displayed		
	Cart can be emptied		
Python file handling are well applied	Cart.csv file is created		
	Cart items written to cart.csv		
	Cart items is retrieved from cart.csv		
	All cart items can be removed from cart.csv		
Python functions are correctly applied	Function to load cart data is created		
	Function for adding items to cart is created		
	Function for removing items to cart is created		
	Function for unpacking items from cart is created		
Python control structure are well applied	Proper conditional statements are used		
	Proper looping statements is used		
Decision			

Solution/output:

```
import csv
import os

CART_FILE = 'cart.csv'

def load_cart():
```

```
"""Load the cart data from the CSV file."""
cart = []
if os.path.exists(CART_FILE):
with open(CART_FILE, mode='r') as file:
reader = csv.reader(file)
cart = list(reader)
return cart

def save_cart(cart):
"""Save the cart data to the CSV file."""
with open(CART_FILE, mode='w', newline='') as file:
writer = csv.writer(file)
writer.writerows(cart)

def add_item(name, quantity):
"""Add an item to the cart."""
cart = load_cart()
cart.append([name, quantity])
save_cart(cart)
print(f'Added {quantity} of {name} to the cart.')

def remove_item(name):
"""Remove an item from the cart."""
cart = load_cart()
cart = [item for item in cart if item[0] != name]
save_cart(cart)
print(f'Removed {name} from the cart.')

def empty_cart():
"""Empty the cart."""
save_cart([])
print('The cart has been emptied.')

def view_cart():
"""Display the items in the cart."""
cart = load_cart()
if not cart:
print('The cart is empty.')
else:
print('Items in the cart:')
for item in cart:
```

```
print('- {item[0]}: {item[1]}')

def main():
"""Main function to handle user commands."""
while True:
print('\nShopping Cart Menu:')
print('1. Add item')
print('2. Remove item')
print('3. View cart')
print('4. Empty cart')
print('5. Quit')

choice = input('Choose an option (1-5): ')
if choice == '1':
name = input('Enter the item name: ')
quantity = input('Enter the quantity: ')
add_item(name, quantity)
elif choice == '2':
name = input('Enter the item name to remove: ')
remove_item(name)
elif choice == '3':
view_cart()
elif choice == '4':
empty_cart()
elif choice == '5':
print('Exiting the program. Goodbye!')
break
else:
print('Invalid choice. Please try again.')

if __name__ == '__main__':
main()
```

END

Further information to the trainer

Hi mana, E., Richard, M., Bajpai, G., Louis, S., & Kayalvizhi, J. (2021). Implementation of IoT Framework with Data Analysis Using Deep Learning Methods for Occupancy Predic on in a Building. Future Internet.

LazyProgrammer. (2016). Deep Learning: Recurrent Neural Networks in Python: LSTM, GRU, and more RNN machine learning architectures in Python and Theano (Machine Learning in Python).

Mar n, K., Hi mana, E., Ngabonziza, J., Hanyurwimfura, D., Musabe, R., Uwamahoro, A.,Mutonga, K. (2023). Crop Yield Prediction Using Machine Learning Models: Case of Irish Potato and Maize. Agriculture, 20.

Moolayil, J. J. (2019). Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python.

Morgan, P. (2018). Data Analysis from Scratch With Python: Beginner Guide, Pandas, NumPy, Scikit-Learn, IPython, TensorFlow, and Matplotlib.

Russell, R. (2018). Machine Learning: Step-by-Step Guide To Implement Machine Learning Algorithms with Python.

Sarkar, D., Raghav, B., & Tushar, S. (2017). Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems.

James, G., Wi en, D., Has e, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: with Applications in R. Springer.

https://www.youtube.com/watch?v=kqtD5dpn9C8

https://www.coursera.org/specializations/python

https://cs50.harvard.edu/python/2022/

https://www.coursera.org/specializations/python

https://docs.python.org/3/

https://automatetheboringstuff.com/

```
#  importing whole modele
   from tkinter import *
   from tkinter .ttr import *

#importing strfitime function to
#retrieve system 's time
from time import strftime

#creating tkinter window
root = Tk()
root.title ('Clock')

#This function is used to
#display time on the label

def time():
        string = strftime('%H :%M:%S %P')
```

Indicative contents

3.1 Applying OOP Concepts
3.2 Applying python Date and time concepts
3.3 Applying Python Libraries
3.4 Applying system Automation

Key Competencies for Learning Outcome 3: Apply Object-Driven in Python

Knowledge	Skills	Attitudes
● Description of date and time ● Description of Python Library ● Understanding Scope of library according to the name space ● Identification of tasks to automate ● Identification of tasks to be prioritized	● Applying OOP Concepts ● Setting Time zones ● Formatting and parsing ● Performing relative timedeltas ● Using python libraries ● Selecting Python Automation Library ● Developing Python Script ● Integrating script with Deployment Process ● Testing and Monitoring the automated task	● Having teamwork spirit ● Being critical thinker ● Being innovative ● Being attentive ● Being creative ● Problem solving ● Being practical oriented

Duration: 40 hrs

Learning outcome 3 objectives:

By the end of the learning outcome, the trainees will be able to:

1. Describe correctly date and time as according to python standards
2. Describe correctly Python Library in accordance with python standards
3. Understand correctly Scope of library according to the name space
4. Identify clearly tasks to automate based on specific task
5. Identify correctly tasks to be prioritized based on specific task
6. Apply correctly OOP Concepts in line with python standards
7. Set properly Time zones in line with python standard
8. Use correctly python libraries in accordance with python standards
9. Select correctly Python Automation Library based on specific task
10. Develop correctly Python Script based on specific task to be automated
11. Integrate properly script with Deployment Process based on specific task
12. Test and monitor correctly automated task in accordance with python standards

Resources

Equipment	**Tools**	**Materials**
• Computer	• Python (latest and stable version) • IDE • Jupyter notebook	• Internet

Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have python installed on all computers to be used
- Have prepared computer lab
- Have IDE (Pycharm) installed on all computers to be used

Indicative content 3.1: Applying OOP Concepts

Duration: 10 hrs

Theoretical Activity 3.1.1: Description of OOP concepts

Notes to the trainer:

- While delivering this activity small groups can be used to discuss on OOP concepts.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to to answer the following questions related to the introduction to Object Oriented Programming.
Describe the concepts as applied in OOP:

i. Object
ii. Class
iii. Inheritance
iv. Polymorphism
v. Encapsulation

Step 2: Ask trainees to write their answers on paper or flipchart

Step 3: Ask trainees to present their findings

Step 4: Provide expert view by using didactic materials

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 3.1.1 in the trainee manual

Points to Remember

- Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects," which are instances of classes that encapsulate data and behaviours.

- Classes serve as blueprints for creating objects, defining attributes and methods.

- Inheritance allows one class (subclass) to inherit properties and methods from another class (superclass), promoting code reuse and establishing a hierarchical relationship.
- Polymorphism enables objects of different classes to be treated as instances of a common superclass, allowing methods to be defined in multiple forms.
- Encapsulation restricts direct access to an object's internal state, exposing only necessary components through public methods, thereby enhancing data protection and modularity.
- Together, these concepts form the foundation of OOP, facilitating organized, reusable, and maintainable code.

Practical Activity 3.1.2: Application of classes and objects

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply classes and objects in python programs.
- While delivering this content the trainer has to:

✓ Avail computer installed with python and PyCharm.

✓ Prepare python program that will be developed including the application of classes and objects.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As a machine learning engineer, you have been asked to develop a python program to perform simple calculation (add, subtract, divide and multiply). That program has to let numbers to be entered by user using keyboard, and after entering numbers it has to let user to select the operation to be performed and display the results according to selected operation.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to apply classes and object in python while developing a simple calculator, while demonstrating explain the code to be used.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 3.1.2.

Points to Remember

- Classes and objects in Python provide a powerful way to model complex systems, encapsulate data, and promote code reuse. From simple data structures to complex frameworks, object-oriented programming enhances the clarity and maintainability of code.
- Classes and object in python are most applicable in different ways like the followings: in the development of calculator programs, in banking transaction programs, and in library management programs.

Practical Activity 3.1.3: Applying inheritance in python

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply inheritance in python programs.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Prepare python program that will be developed including the application of inheritance.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you have been asked to develop a python program to perform bank transactions including deposit and withdraw, for withdraw you cannot let the account to be empty means it has to let 100 Rfw on the account and for all transactions it has to display the messages.
Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to apply inheritance in python while developing a simple bank transaction program, while demonstrating explain the line of code to be used.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 3.1.3.

Points to Remember

- Inheritance in Python is a powerful feature that allows one class (the subclass) to inherit attributes and methods from another class (the superclass).
- Common applications of inheritance in Python are: Code reusability, method overriding, creating a hierarchical structure, multiple inheritance, and framework and library development.

Practical Activity 3.1.4: Applying polymorphism in python

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply polymorphism in python programs.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Prepare python program that will be developed including the application of polymorphism.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you have been asked to develop a python program to perform bank transactions including deposit and withdraw, for withdraw you cannot let the account to be empty; means it has to let 100 Rfw on the account

and for all transactions it have to display the messages the program have to include the application of polymorphism.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to apply polymorphism in python while developing a simple bank transaction program, while demonstrating explain the line of code to be used.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 3.1.4.

Points to Remember

- Polymorphism in Python is a powerful feature that allows different classes to be treated as instances of the same class through a common interface.
- Polymorphism can be applied in different ways including: Method overriding, duck typing, operator overloading, function overloading, and frameworks and libraries.

Practical Activity 3.1.5: Applying Encapsulation

Notes to the trainer:

- This activity has to take place in computer lab where trainees have to apply polymorphism in python programs.
- While delivering this content the trainer has to:
  - ✓ Avail computer installed with python and PyCharm.
  - ✓ Prepare python program that will be developed including the application of encapsulation.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you have been asked to develop a python program to perform simple calculation (add, subtract, divide and multiply). That program has to

let numbers to be entered by user using keyboard, and after entering numbers it has to let user to select the operation to be performed and display the results according to selected operation. In addition, the program has to show the application of encapsulation.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to apply classes and object in python while developing a simple calculator, while demonstrating explain the code to be used.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 3.1.5.

Points to Remember

- Encapsulation in Python is a fundamental object-oriented programming principle that restricts direct access to certain attributes and methods of a class.
- Encapsulation can be used in different ways including: Data hiding, controlled access, implementation hiding, API design, and security.

Application of learning 3.1.

As machine learning engineer, you are requested to develop a simple bank account transaction python program where there is application of inheritance the system have to accept to input the account owner and choose the action to be performed including withdraw, deposit and the system have to show the amount to be withdrawn based on settings of bank. The program has to include the followings: Object, class, inheritance, polymorphism, and encapsulation.

Checklist:

Criteria	Indicators	Observation	
		Yes	No
Classes and objects are well applied	Base class for a bank account is declared		
	The balance is initialised to zero		
	Initialize the account with owner and starting balance		
	Add amount to the balance is performed		
	Calculate the maximum amount that can be withdrawn is performed		
	Creating an object of the appropriate account class		
Inheritance is well applied	Derived class for a savings account is used		
	Initialize the savings account with owner, balance, and interest rate		
	Apply interest to the current balance is performed		
	Derived class for a checking account is used		
	Derived class for a checking account is used		
	Calculate the maximum amount that can be withdrawn is performed		
	Allow withdrawal up to the overdraft limit is done		
Polymorphism is well applied	User input for account details is performed		
	Ensure the initial balance is above 100		
	User operations is displayed		
	Choose account type is displayed		
Decision			

Solution/output:

Python program for simple bank account transaction:

```
# Base class for a bank account

class BankAccount:

def __init__(self, owner, balance=0):

# Initialize the account with owner and starting balance
```

```
self.owner = owner
self.balance = balance
def deposit(self, amount):
# Add amount to the balance
self.balance += amount
return f"Deposited {amount}. New balance: {self.balance}"
def withdraw(self, amount):
# Calculate the maximum amount that can be withdrawn
if self.balance - amount < 100:
max_withdrawable = self.balance - 100
return f"You can only withdraw up to {max_withdrawable}."
self.balance -= amount
return f"Withdrew {amount}. New balance: {self.balance}"
# Derived class for a savings account
class SavingsAccount(BankAccount):
def __init__(self, owner, balance=0, interest_rate=0.02):
# Initialize the savings account with owner, balance, and interest rate
super().__init__(owner, balance)  # Call the constructor of the base class
self.interest_rate = interest_rate
def apply_interest(self):
# Apply interest to the current balance
interest = self.balance * self.interest_rate
self.balance += interest
return f"Interest applied. New balance: {self.balance}"
# Derived class for a checking account
class CheckingAccount(BankAccount):
def __init__(self, owner, balance=0, overdraft_limit=100):
# Initialize the checking account with owner, balance, and overdraft limit
```

```
super().__init__(owner, balance)
self.overdraft_limit = overdraft_limit
def withdraw(self, amount):
# Calculate the maximum amount that can be withdrawn
if self.balance - amount < 100:
max_withdrawable = self.balance - 100
return f"You can only withdraw up to {max_withdrawable}."
# Allow withdrawal up to the overdraft limit
if amount > self.balance + self.overdraft_limit:
return "Insufficient funds, even with overdraft."
self.balance -= amount
return f"Withdrew {amount}. New balance: {self.balance}"
# User input for account details
owner = input("Enter the account owner's name: ")
initial_balance = float(input("Enter the initial balance: "))
# Ensure the initial balance is above 100
while initial_balance <= 100:
print("Initial balance must be greater than 100.")
initial_balance = float(input("Enter the initial balance: "))
# Choose account type
account_type = input("Enter account type (savings/checking): ").lower()
# Creating an object of the appropriate account class
if account_type == 'savings':
account = SavingsAccount(owner, initial_balance)
elif account_type == 'checking':
account = CheckingAccount(owner, initial_balance)
else:
print("Invalid account type. Defaulting to BankAccount.")
```

```
account = BankAccount(owner, initial_balance)

# User operations

while True:

action = input("Do you want to deposit, withdraw, or apply interest? (d/w/i/q): ").lower()

if action == 'd':

amount = float(input("Enter amount to deposit: "))

print(account.deposit(amount))

elif action == 'w':

amount = float(input("Enter amount to withdraw: "))

print(account.withdraw(amount))

elif action == 'i' and isinstance(account, SavingsAccount):

print(account.apply_interest())

elif action == 'q':

break

else:

print("Invalid option. Please try again.")
```

Indicative content 3.2: Applying python Date and time concepts

Duration: 10 hrs

Theoretical Activity 3.2.1: Description of date and time

Notes to the trainer:

- While delivering this content, small groups can be used for describing date and time in python.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to describe date and time concepts in python programming:

i. Datetime
ii. Dateutil
iii. Arrow
iv. Pendulum
v. Python-tzdata

Step 2: Ask trainees to write their answers on paper or flipchart

Step 3: Ask trainees to present their findings

Step 4: Provide expert view by using didactic materials

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 3.2.1 in the trainee manual

Points to Remember

- Python provides robust support for date and time manipulation through its built-in datetime module and several powerful third-party libraries like dateutil, Arrow, Pendulum, and python-tzdata.
- The datetime module is part of the standard library and provides classes for manipulating dates and times. It includes functionalities for creating, formatting, and performing arithmetic on dates and times.
- Dateutil is a powerful extension of the datetime module that provides additional features, such as parsing dates from strings and handling time zones. It simplifies date manipulation with utilities for relative deltas, recurrence rules, and more.
- Arrow is a lightweight library that simplifies working with dates and times in Python. It provides an intuitive API for creating, formatting, and converting dates, along with built-in timezone handling and human-friendly features.
- Pendulum is a robust datetime library that extends datetime with advanced features like duration calculations, timezone conversions, and natural language support. It offers immutable instances, making it easier to work with dates and times without side effects.
- python-tzdata is a package that provides the IANA time zone database for Python applications. It allows for accurate timezone conversions and offsets, ensuring applications handle date and time correctly across different regions.

Practical Activity 3.2.2: Applying python Date and time concepts

Notes to the trainer

- This activity should take place in computer lab where trainees have to apply python date and time concepts.
- While delivering this content, you are required to:

✓ Avail python and PyCharm installed on all computer to be used.
✓ Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab apply date and time concepts while performing different operations on date and time calculating the difference between date, calculating the age depending on current date and date of birth.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to apply date and time concepts. And while demonstrating explain the application of each concept.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 3.2.2.

Points to Remember

- In Python, you can effectively manage time zones using pytz, format and parse dates using strftime and strptime, and perform arithmetic with dates using timedelta and relativedelta.
- These tools provide a robust framework for handling date and time calculations in your applications.

Application of learning 3.2.

As machine learning engineer, you are asked to develop a python program that can be used while calculating the age of students in order to know if they are allowed to take national id card depending on entered age and the current date.

The program has to tell user if the entered student is allowed or not.

Checklist:

Criteria	Indicators	Observation	
		Yes	No
Python Date and time concepts are well applied	Datetime library is imported		
	Get user input for birth date is done		
	Calculate age is performed		
	Adjust age if the birthday hasn't occurred yet this year is done		
	Check if allowed for ID is done		
Decision			

Solution/output:

```
from datetime import datetime
print("Welcome to the Student Age Verification System")
# Get user input for birth date
birth_date = input("Please enter the student's birth date (YYYY-MM-DD): ")
try:
# Calculate age
birth_date = datetime.strptime(birth_date, "%Y-%m-%d")
today = datetime.today()
age = today.year - birth_date.year
# Adjust age if the birthday hasn't occurred yet this year
if (today.month, today.day) < (birth_date.month, birth_date.day):
age -= 1
# Check if allowed for ID
allowed = age >= 18
if allowed:
print(f"The student is {age} years old and is allowed to take a national ID card.")
else:
print(f"The student is {age} years old and is NOT allowed to take a national ID card.")
except ValueError:
print("Invalid date format. Please enter the date in the format 'YYYY-MM-DD'.")
```

Explanation

- The program first prompts for the birth date and attempts to parse it.
- It calculates the age directly within the main block and checks the eligibility for a national ID card.
- Finally, it prints the result or an error message if the input format is invalid.

Indicative content 3.3: Applying Python Libraries

Duration: 10 hrs

Theoretical Activity 3.3.1: Description of Python Libraries

Notes to the trainer:

- While delivering this content, small groups can be used for describing python libraries.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to describe the following python libraries:

i. Matplotlib

ii. Numpy

iii. Pandas

Step 2: Ask trainees to write their answers on paper or flipchart

Step 3: Ask trainees to present their findings

Step 4: Provide expert view by using didactic materials

Step 5: Address any questions or concerns

Step 6: Ask trainees to read the key reading 3.3.1 in the trainee manual

Points to Remember

- The Python Standard Library provides a robust foundation for programming tasks, while libraries like Matplotlib, NumPy, and Pandas significantly enhance Python's capabilities in data visualization, numerical computing, and data analysis.
- Together, these libraries make Python a powerful tool for scientific computing, data analysis, and more.

Practical Activity 3.3.2: Using python libraries

Notes to the trainer:

- This activity should take place in computer lab where trainees have to use python libraries.
- While delivering this content, you are required to:

    i. Avail python and PyCharm IDE installed on all computer to be used.
    ii. Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab to use python libraries described in key readings 3.3.1.
Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to use python libraries and while demonstrating explain how to use python libraries.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 3.3.2.

Points to Remember

- To use any library in Python, you need to import it. This is typically done at the beginning of your script or notebook.
- NumPy provides a powerful array object and a range of functions for numerical operations.
- Pandas is used for data manipulation and analysis, primarily with its DataFrame structure.
- Matplotlib is used for creating static, animated, and interactive visualizations.

Application of learning 3.3.

You’re working for a retail store located in Nyanza District. You are tasked with analysing the monthly sales data for a retail store. The sales data is manually defined in a Python dictionary. The objective is to calculate total and average sales, and visualize the sales trend over the months using Python libraries.

Checklist:

Criteria	**Indicators**	**Observation**	
		Yes	**No**
Python libraries are well used	Pandas library is installed		
	Mathplotlib is installed		
	Sample sales data are displayed		
	DataFrame is created		
	Total and average sales are calculated		
	Visualize sales trend is performed		
Decision			

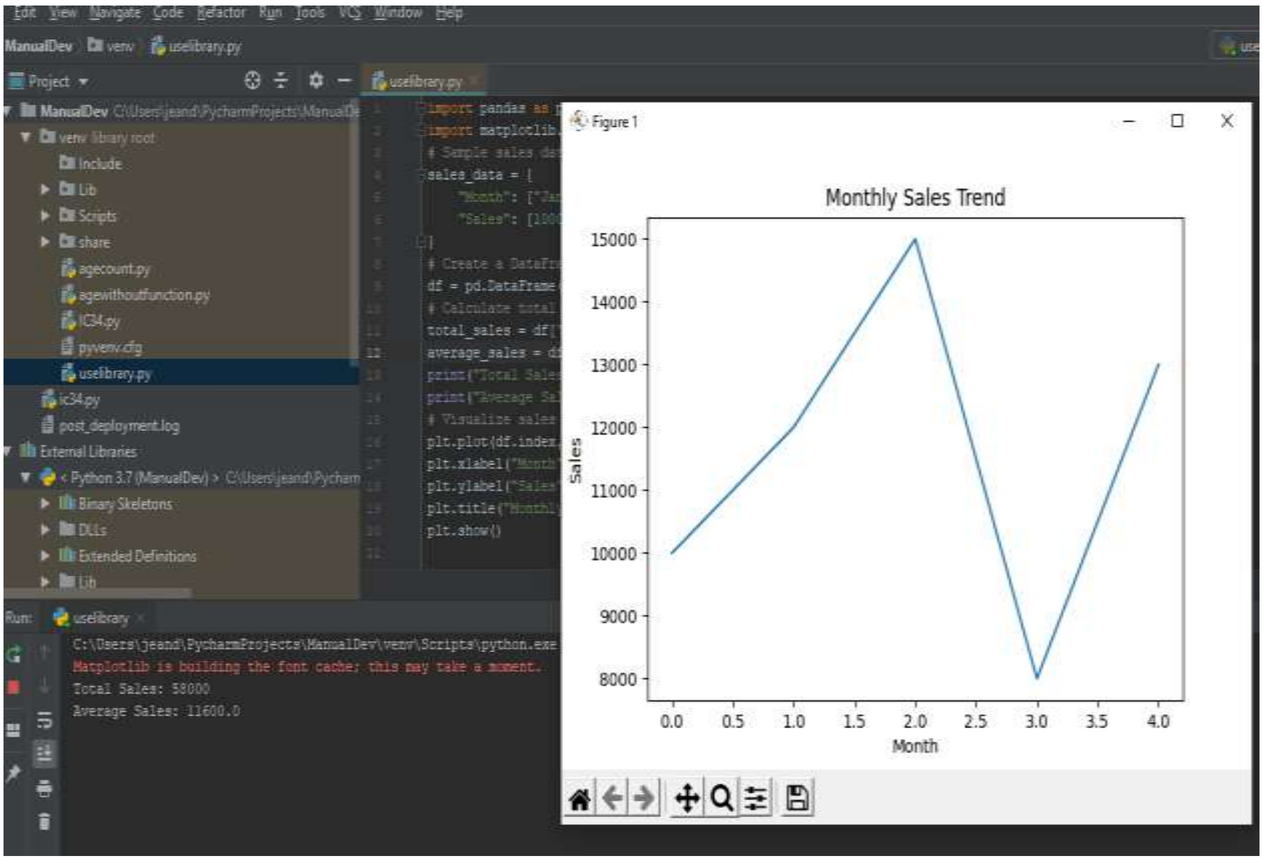
Solution/Output:

```
import pandas as pd
import matplotlib.pyplot as plt
# Sample sales data (replace with your actual data)
sales_data = {
"Month": ["Jan", "Feb", "Mar", "Apr", "May"],
"Sales": [10000, 12000, 15000, 8000, 13000]}
# Create a DataFrame
df = pd.DataFrame(sales_data)
# Calculate total and average sales
total_sales = df["Sales"].sum()
average_sales = df["Sales"].mean()
print("Total Sales:", total_sales)
print("Average Sales:", average_sales)
# Visualize sales trend
plt.plot(df.index, df["Sales"])
plt.xlabel("Month")
plt.ylabel("Sales")
plt.title("Monthly Sales Trend")
plt.show()
```

Explaination:

- The pandas library is used to create a DataFrame from the sales data dictionary.
- The sum() and mean() functions are used to calculate total and average sales.
- Matplotlib is used to create a line chart to visualize the sales trend over the months.

Output:



```
import pandas as p
import matplotlib.
# Sample sales da
sales_data = {
    "Month": ["Ja
    "Sales": [100
}
# Create a DataFr
df = pd.DataFrame
# Calculate total
total_sales = df[
average_sales = d
print("Total Sale
print("Average Sa
# Visualize sales
plt.plot(df.index
plt.xlabel("Month
plt.ylabel("Sales
plt.title("Monthl
plt.show()
```

```
C:\Users\jeand\PycharmProjects\ManualDev\venv\Scripts\python.exe
Matplotlib is building the font cache; this may take a moment.
Total Sales: 58000
Average Sales: 11600.0
```

Indicative content 3.4: Applying system Automation

Duration: 10 hrs

Theoretical Activity 3.4.1: Identification of tasks to automate and to be prioritized

Notes to the trainer:

- While delivering this content, small groups can be used to identify task to automate and to be prioritised.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to identify tasks to automate and to be prioritized.
Step 2: Ask trainees to write their answers on paper or flipchart
Step 3: Ask trainees to present their findings
Step 4: Provide expert view by using didactic materials
Step 5: Address any questions or concerns
Step 6: Ask trainees to read the key reading 3.4.1 in the trainee manual

Points to Remember

- Automating certain tasks can greatly improve efficiency, reduce errors, and speed up deployment processes.
- By focusing on repetitive, time-consuming, error-prone tasks that are critical for deployment speed, teams can maximize their productivity and ensure a smoother workflow.
- Automating these tasks not only saves time but also enhances the reliability of the development and deployment processes.

Practical Activity 3.4.2: Installing Python Automation Libraries

Notes to the trainer:

- This activity should take place in computer lab where trainees have to select python automation library.
- While delivering this content, you are required to:
  - ✓ Avail python and PyCharm IDE installed on all computer to be used.

  Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:

As machine learning engineer, you are asked to go to the computer lab to install python automation libraries.:

i. fabric
ii. ansible
iii. salt
iv. boto3
v. vsphere-automation-sdk

Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to install python automation libraries, while demonstrating explain the steps to be followed.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 3.4.2.

Points to Remember

- To install python automation libraries, you can use pip install library name and remember to replace the library name with the one that you are installing like: pip install fabric, pip install ansible, pip install salt, pip install boto3, pip install vSphere-automation-sdk.

Practical Activity 3.4.3: Developing python script

Notes to the trainer:

- This activity should take place in computer lab where trainees have to develop python script.
- While delivering this content, you are required to:
  - ✓ Avail python and PyCharm IDE installed on all computer to be used.
  - ✓ Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As machine learning engineer, you are asked to go to the computer lab to develop python scripts.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to develop python scripts, while demonstrating explain the steps to be followed.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 3.4.3.

Points to Remember

- To develop Python scripts effectively, plan your goals, import necessary libraries, structure your code logically, write functions for reusable code, use library functions, handle errors, and document your work.
- Consider using a virtual environment to isolate dependencies and leverage logging for debugging.

- By following these steps and incorporating best practices, you can create efficient, maintainable, and well-structured Python scripts

Run the Tests

Practical Activity 3.4.4: Integrating script with Deployment process

Notes to the trainer:

- This activity should take place in computer lab where trainees have to integrate script with deployment process.
- While delivering this content, you are required to:
  - ✓ Avail python and PyCharm IDE installed on all computer to be used.
  - ✓ Avail sample deployed application to be used.
  - ✓ Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As a machine learning engineer, you are asked to go to the computer lab to integrate script with deployment process.

Step 2: Explain the task and provide clear work instruction.

Step 3: Demonstrate how to integrate script with deployment process, while demonstrating explain the steps to be followed.

Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.

Step 5: Verify whether the tasks are clearly performed.

Step 6: Provide feedback if necessary.

Step 7: Ask trainees to read key reading 3.4.4.

Points to Remember

- While Integrating script with Deployment Process you follow the flowing steps:

  Step 1: Trigger method to initiate the Python script post-deployment

  Step 2: Direct execution after deployment completion

Step 3: Integration with CI/CD pipelines
Step 4: Scheduled execution at specific intervals
Step 5: Implement security measures to restrict script access and control sensitive information.

Practical Activity 3.4.5: Testing and monitoring the automated tasks

Notes to the trainer

- This activity should take place in computer lab where trainees have to test and monitor the automated tasks.
- While delivering this content, you are required to:
  - ✓ Avail python and PyCharm IDE installed on all computer to be used.
  - ✓ Avail sample deployed application to be used.
  - ✓ Avail video as didactic material.

Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the topic and ask trainees to do the task described below:
As a machine learning engineer, you are asked to go to the computer lab to test and monitor the automated tasks.
Step 2: Explain the task and provide clear work instruction.
Step 3: Demonstrate how to test and monitor the automated tasks, while demonstrating explain the steps to be followed.
Step 4: Asks trainees to perform the activity of step 3 and monitor the procedures.
Step 5: Verify whether the tasks are clearly performed.
Step 6: Provide feedback if necessary.
Step 7: Ask trainees to read key reading 3.4.5.

Points to Remember

- While testing and monitoring the automated task you have to: Thorough testing, monitor script logs execution, and refine and improve.

Application of learning 3.4.

Manzi as a machine learning engineer, has built a web application for BERWA School. After each deployment of the application, certain tasks must be automated to ensure the application is correctly configured and ready for use. These tasks include configuring the server, backing up the database, and sending notifications upon completion. To enhance automation, Manzi decides to integrate a Python script into the deployment process, ensuring it runs after each successful deployment and meets necessary security measures. As a machine learning engineer, you are asked to integrate a Python script into the deployment process to handle these tasks automatically, ensuring they are secure and run only after a successful deployment.

Checklist:

Criteria	Indicators	Observation	
		Yes	No
Python Automation Library are well installed	boto3 is installed		
	import necessary packages and library is done		
Python script is developed	Setup logging is performed		
	Function to get secrets from AWS Secrets Manager is developed		
	Function to configure server is developed		
	Set environment variable for app mode is done		
	Function to clean up temporary files is developed		
	Function to trigger database backup is developed		
	Function to send notification (e.g., via email or API) is developed		
	Main function to run post-deployment tasks is developed		
Decision			

Solution/output:

```
import os
import subprocess
import shutil
import logging
from datetime import datetime
import boto3

# Setup logging
logging.basicConfig(filename='post_deployment.log', level=logging.INFO,
format='%(asctime)s - %(levelname)s - %(message)s')
# Function to get secrets from AWS Secrets Manager
def get_secret(secret_id):
try:
client = boto3.client('secretsmanager')
secret_value = client.get_secret_value(SecretId=secret_id)
logging.info(f"Successfully retrieved secret {secret_id}")
return secret_value['SecretString']
except Exception as e:
logging.error(f"Failed to retrieve secret {secret_id}: {e}")
raise
# Function to configure server
def configure_server():
try:
# Set environment variable for app mode (example)
os.environ['APP_MODE'] = 'production'
logging.info(f"Server configured to {os.getenv('APP_MODE')} mode.")
except Exception as e:
logging.error(f"Failed to configure server: {e}")
raise
# Function to clean up temporary files
def clean_temp_files():
try:
temp_dir = '/tmp/app_temp/'
if os.path.exists(temp_dir):
shutil.rmtree(temp_dir)
logging.info(f"Temporary files in {temp_dir} cleaned.")
except Exception as e:
logging.error(f"Failed to clean temporary files: {e}")
raise
```

```
# Function to trigger database backup
def trigger_backup():
try:
# Get database password from environment variables or secret manager
db_password = os.getenv('DB_PASSWORD') or get_secret('DB_PASSWORD_SECRET_ID')
# Example of triggering a backup command (pseudo-command)
backup_command = f"backup_command --db production_db –password {db_password}"
subprocess.run(backup_command.split(), check=True)
logging.info("Database backup completed.")
except subprocess.CalledProcessError as e:
logging.error(f"Backup command failed: {e}")
raise
except Exception as e:
logging.error(f"Failed to trigger backup: {e}")
raise
# Function to send notification (e.g., via email or API)
def send_notification(message):
try:
# Placeholder for sending notification (pseudo-code)
# For example, send an email or API request to a notification service
logging.info(f"Notification sent: {message}")
except Exception as e:
logging.error(f"Failed to send notification: {e}")
raise
# Main function to run post-deployment tasks
def main():
try:
logging.info("Starting post-deployment tasks.")
configure_server()  # Configure the server environment
clean_temp_files()  # Clean up temporary files
trigger_backup()    # Trigger the database backup
send_notification("Post-deployment tasks completed successfully.")   # Send success
notification
logging.info("All post-deployment tasks completed successfully.")
except Exception as e:
logging.error(f"Post-deployment tasks failed: {e}")
send_notification(f"Post-deployment tasks failed: {e}")
raise
if __name__ == "__main__":
main()
```

Learning outcome 3 end assessment

Written assessment

I. Match the following terms with their corresponding definitions as applied in functions:

ANSWER	**Items**	**Definitions**
1……….	1. Class	A. A blueprint for creating objects.
2……….	2. Object	B. A specific instance of a class.
3……….	3. Inheritance	C. The ability to use a method in different ways.
4……….	4. Polymorphism	D. Restricting access to certain components of an object.
5……….	5. Encapsulation	E. A class from which another class inherits.
6……….	6. Method Overriding	F. A derived class that inherits properties from another class.
7……….	7. Constructor	G. A method that replaces the implementation of a method in the superclass.
8……….	8. Data Hiding	H. A special method used to initialize objects.
9……….	9. Superclass	I. The ability to hide data from outside access.
10……….	10. Subclass	J. The mechanism of a class acquiring properties from another class.

ANSWER:

1. Class - A
2. Object - B
3. Inheritance - J
4. Polymorphism - C
5. Encapsulation - D
6. Method Overriding - G
7. Constructor - H
8. Data Hiding - I
9. Superclass - E
10. Subclass – F

II. **Match the following Python libraries with their corresponding primary use cases:**

ANSWER	Python libraries	Primary use cases
1……….	1. Matplotlib	A. Data manipulation and analysis
2……….	2. NumPy	B. Numerical operations on arrays and matrices
3……….	3. Pandas	C. Data visualization
4……….	4. Ansible	D. Automation and configuration management
5……….	5. datetime	E. Date and time handling

ANSWER:

1. Matplotlib - C
2. NumPy - B
3. Pandas - A
4. Ansible - D
5. Datetime - E

III. Circle the letter corresponding to the correct answer :

1. What keyword is used to define a class in Python?
   A) define
   B) class
   C) object
   D) function

   Answer: B) class

2. Which of the following allows a class to inherit properties from multiple classes?
   A) Single Inheritance
   B) Multiple Inheritance
   C) Multilevel Inheritance
   D) Hierarchical Inheritance

   ANSWER: B) Multiple Inheritance

3. What does the __init__() method do?
   A) It creates a new class.
   B) It initializes an object's attributes.
   C) It defines a new method.
   D) It overrides a method.

ANSWERS: B) It initializes an object's attributes.

4. What is the primary benefit of encapsulation?
   A) It increases redundancy.
   B) It hides the internal state of an object.
   C) It allows multiple inheritance.
   D) It simplifies code readability.

ANSWER:B) It hides the internal state of an object.

5. In Python, what is polymorphism primarily used for?
   A) To create new classes.
   B) To allow different classes to be treated as instances of the same class.
   C) To hide data.
   D) To define class methods.

ANSWER:B) To allow different classes to be treated as instances of the same class.

6. Which Python library is most commonly used for working with dates and times?
   A) NumPy
   B) Pandas
   C) datetime
   D) Matplotlib

ANSWER: C) datetime

7. What is the purpose of the timezone attribute in the datetime module?
   A) To convert dates and times between different time zones
   B) To set the current time zone for the system
   C) To calculate time differences between time zones
   D) To format dates and times according to a specific time zone

ANSWER: A) To convert dates and times between different time zones

8. Which Python library is specifically designed for working with time zones?
   A) datetime
   B) dateutil
   C) Arrow
   D) Pendulum

ANSWER: B) dateutil

9. What is the correct syntax for creating a datetime object representing the current time?
   A) datetime.now()

B) time.time()
C) calendar.time()
D) date.today()

ANSWER: A) datetime.now()

10. Which Python library is commonly used for data visualization and plotting?
    A) NumPy
    B) Pandas
    C) Matplotlib
    D) datetime

ANSWER: C) Matplotlib

IV. **State whether the following statements are True or False**.

1. An object is an instance of a class.
2. In Python, all classes must inherit from a superclass.
3. Encapsulation allows for direct access to an object's private attributes.
4. Polymorphism can be achieved through method overloading.
5. A subclass can override methods from its superclass.
6. Inheritance promotes code reuse.
7. The self keyword is used to refer to an instance of a class.
8. All attributes in a class are public by default.
9. The __str__() method is used to represent an object as a string.
10. Method overloading is directly supported in Python.
11. The datetime module provides functions for formatting and parsing dates and times.
12. The dateutil library is primarily used for numerical operations on arrays and matrices.
13. The arrow library offers a more intuitive API for working with dates and times.
14. The Pendulum library is specifically designed for time zone handling.
15. The Python-tzdata library is included by default in Python installations.

ANSWER:

1. True
2. False
3. False
4. False
5. True
6. True
7. True
8. True
9. True
10. False

11. True
12. False
13. True
14. False
15. False

V. Complete the followings with correct keyword from the followings

class, inheritance, class, constructor, Method overriding, encapsulation, object, hierarchical, polymorphism, Encapsulation, strftime() ,schedule, Ansible, timedelta() ,Pandas

1. A __________ is a blueprint for creating objects in Python.
2. The process of a class inheriting properties from another class is called __________.
3. In Python, you use the keyword __________ to define a class.
4. The __________ method is automatically called when an object is created.
5. __________ allows a subclass to provide a specific implementation of a method defined in its superclass.
6. Data hiding is a feature of __________ that restricts direct access to some attributes.
7. A __________ is an instance of a class.
8. Inheritance allows for the creation of a __________ structure among classes.
9. The ability to treat objects of different classes as objects of a common superclass is known as __________.
10. __________ is achieved when a subclass inherits methods and properties from a superclass.
11. To convert a datetime object to a string, you can use the _______ method.
12. The _______ method can be used to calculate the difference between two datetime objects.
13. The _______ library provides tools for automating tasks related to system administration.
14. To schedule a task to run at a specific time, you can use the _______ module.
15. The _______ library is commonly used for working with tabular data.

ANSWER:

1. class
2. inheritance.
3. class
4. constructor
5. Method overriding
6. encapsulation
7. object
8. hierarchical
9. polymorphism

10. Encapsulation
11. strftime()
12. timedelta()
13. Ansible
14. schedule
15. Pandas

Practical assessment

As a machine learning engineer, you have been assigned the task of developing and managing a Health Management System for GIRUBUZIMA HOSPITAL. The system will handle patient data, automate deployment tasks, and analyze health trends to ensure smooth operation and efficient service delivery. This project involves applying concepts from Object-Oriented Programming (OOP), date and time handling, Python libraries, and automation of post-deployment tasks.

Tasks:

1. Create a Patient class with attributes like name, age, gender, and disease.
2. Create specialized classes like In_patient and Outpatient inheriting from Patient.
3. Use methods that behave differently based on patient type.
4. Keep patient information private and control access through methods.
5. Ensure records are aligned with **Rwanda's timezone** (Central Africa Time).
6. Format timestamps for patient records.
7. Calculate days between tests and monitor recovery progress.
8. Visualize patient recovery trends.
9. Perform statistical operations on patient records.
10. Manage large datasets with patient info.
11. **Use Fabric or Ansible** to automate deployment-related tasks such as **restarting services** and **updating configurations**.

Checklist:

Criteria	Indicators	Observation	
		Yes	No
OOP concepts are well applied	Patient class with attributes like name, age, gender, and disease are created		
	Specialized classes like In_patient and Outpatient inheriting from Patient are created		
	Methods that behave differently based on patient type are used		

	Patient information private and control access through methods are kept secret		
Python Date and time concepts are well applied	Records are aligned with Rwanda’s timezone		
	Timestamps for patient records are formated		
	Days between tests and monitor recovery progress are calculated		
	Large datasets with patient info are managed		
Python Libraries are well applied	Pytz library is installed		
	Datetime library is imported		
	Pandas library is installed		
System Automation is well applied	Fabric or ansible library is installed		
	The script is written depending on task to automate		
Decision			

Solution:

```
from datetime import datetime
import pytz


# Base Patient class
class Patient:
def __init__(self, name: str, age: int, gender: str, disease: str):
self.__name = name
self.__age = age
self.__gender = gender
self.__disease = disease
self.__recorded_at = datetime.now(pytz.timezone('Africa/Kigali'))

def get_patient_info(self):
return {
"name": self.__name,
"age": self.__age,
"gender": self.__gender,
"disease": self.__disease,
"recorded_at": self.__recorded_at.strftime("%Y-%m-%d %H:%M:%S")}


# InPatient class
class InPatient(Patient):
def __init__(self, name: str, age: int, gender: str, disease: str, room_number: str,
admission_date: datetime):
super().__init__(name, age, gender, disease)
self.room_number = room_number
self.admission_date = admission_date

def get_patient_info(self):
info = super().get_patient_info()
info['room_number'] = self.room_number
info['admission_date'] = self.admission_date.strftime("%Y-%m-%d %H:%M:%S")
return info

def calculate_days_hospitalized(self, discharge_date: datetime):
return (discharge_date - self.admission_date).days

def calculate_payment(self, daily_rate: float, discharge_date: datetime):
days_hospitalized = self.calculate_days_hospitalized(discharge_date)
return days_hospitalized * daily_rate

# OutPatient class
class OutPatient(Patient):
def __init__(self, name: str, age: int, gender: str, disease: str, visit_date: datetime):
super().__init__(name, age, gender, disease)
self.visit_date = visit_date

def get_patient_info(self):
info = super().get_patient_info()
info['visit_date'] = self.visit_date.strftime("%Y-%m-%d %H:%M:%S")
return info

# Function to manage patient information
def manage_patients():
# Get patient details from user
name = input("Enter the patient's name: ")
age = int(input("Enter the patient's age: "))
gender = input("Enter the patient's gender: ")
disease = input("Enter the patient's disease: ")

# Admission details
room_number = input("Enter room number (for inpatient): ")
admission_date_str = input("Enter admission date (YYYY-MM-DD): ")
admission_date = datetime.strptime(admission_date_str, "%Y-%m-
%d").replace(tzinfo=pytz.timezone('Africa/Kigali'))
# Create InPatient object
inpatient = InPatient(name, age, gender, disease, room_number, admission_date)

# Discharge details
discharge_date_str = input("Enter discharge date (YYYY-MM-DD): ")
discharge_date = datetime.strptime(discharge_date_str, "%Y-%m-
%d").replace(tzinfo=pytz.timezone('Africa/Kigali'))
```

```
# Daily rate example
daily_rate = float(input("Enter daily rate: "))

# Calculate payment
total_payment = inpatient.calculate_payment(daily_rate, discharge_date)

# Display patient info and payment
print("\nPatient Info:")
print(inpatient.get_patient_info())
print(f"Days hospitalized: {inpatient.calculate_days_hospitalized(discharge_date)}")
print(f"Total payment: ${total_payment:.2f}")

# Example usage
if __name__ == "__main__":
manage_patients()
```

Further information to the trainer

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning (Adaptive Computation and Machine Learning series). MIT Press.
Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.
Has e, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
Chollet, F. (2017). Deep Learning with Python. Manning Publications.
Alpaydin, E. (2020). Introduction to Machine Learning (Adaptive Computation and Machine Learning series). MIT Press.
Bishop, C. Ms. (2006). Pa ern Recognition and Machine Learning. Springer.
https://www.youtube.com/watch?v=kqtD5dpn9C8

https://www.coursera.org/specializations/python

https://cs50.harvard.edu/python/2022/

https://www.coursera.org/specializations/python

https://docs.python.org/3/

https://automatetheboringstuff.com/

October, 2024