

Object Oriented Programming using Python

Codetantra

A training report

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology

(Computer Science and Engineering)

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB



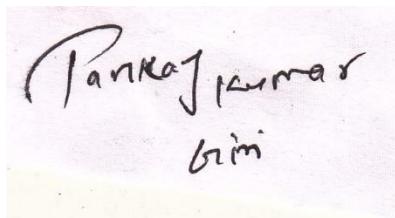
From 01/06/23 to 15/07/23

SUBMITTED BY

Name of Student: Pankaj Giri

Registration Number: 12204745

Signature of the student:



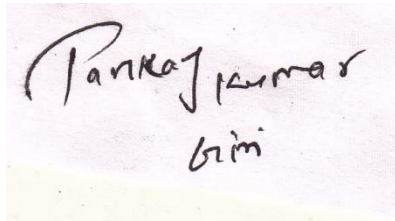
Pankaj Kumar
Giri

To whom so ever it may concern

I, **Pankaj Giri, 12204745**, hereby declare that the work done by me on “**Object Oriented Programming using Python**” from **June, 2023** to **July, 2023**, is a record of original work for the partial fulfillment of the requirements for the award of the degree, **Bachelor of Technology Computer Science and Engineering**.

Pankaj Giri (12204745)

Signature of the student

A photograph of a handwritten signature in black ink on a light-colored piece of paper. The signature is written in a cursive style and reads "Pankaj Kumar" on the first line and "Giri" on the second line.

Dated: **July 15th, 2023**

Summer Training certificate



ACKNOWLEDGEMENT

I would like to express my gratitude towards my university as well as Codetantra for providing me the golden opportunity to do this wonderful summer training regarding Python, which also helped me doing a lot of homework and learning. As a result, I came to know about so many new things. So, I am thankful to them.

Moreover, I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course. I am thankful to have such a good support of them as they always have my back whenever I need.

Also, I would like to mention the support system and consideration of my parents who have always been there in my life to make me choose right thing and oppose the wrong. Without them I could never had learned and became a person who I am now. I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

Date : 15/07/2023

Pankaj Giri

(12204745)

Table of Content

S. No.	Title	Page no
1	Cover Page	1
2	Declaration of the student	2
3	Training Certification from organization	3
4	Acknowledgement	4
5	Table of Contents	5
6	Introduction	6-7
7	Project	8-10
8	Technology learnt	11-14
9	Learning Outcome	15
10	Conclusion	16
11	Bibliography	17

INTRODUCTION

CodeTantra, founded in 2010 by Dr. Sridhar P. Nerur, is a pioneering online platform that offers dynamic coding education. Focused on practical learning, it provides interactive coding exercises, real-world projects, and instant feedback, fostering a comprehensive understanding of programming concepts. With a range of programming languages and collaborative features, CodeTantra empowers learners to bridge theory and practice. Its impact extends to educational institutions, enhancing computer science curricula. Since its inception, CodeTantra continues to shape the coding landscape, equipping individuals with essential skills for the evolving world of software development.

Python Language: Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python: Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like

Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Features in Python:

1. **Easy to code:** Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer-friendly language.
2. **Free and Open Source:** Python language is freely available at the official website. Since it is open source, this means that source code is also available to the public. So you can download it as, use it as well as share it.
3. **Object-Oriented Language:** One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.
4. **GUI Programming Support:** Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.
5. **High-Level Language:** Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.
6. **Extensible feature:** Python is an Extensible language. We can write some Python code into C or C++ language and also, we can compile that code in C/C++ language.
7. **Python is Portable language:** Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

Project:

The project Employee Management System (EMS) is a software application designed to streamline and automate various processes related to employee administration and human resource management within an organization. It serves as a central hub for HR-related tasks, data, and activities, making it easier for businesses to manage their workforce effectively. Here's an overview of the components and features typically found in an Employee Management System:

1. **Employee Class:**

- This class represents an employee. It has attributes like **emp_id**, **name**, **position**, and **salary**.
- The **__init__** method initializes these attributes when an **Employee** object is created.
- The **__str__** method is defined to provide a formatted string representation of an employee's information.

2. **EmployeeManagementSystem Class:**

- This class handles the management of employee data using file handling.
- The **__init__** method takes the name of the file as an argument and stores it in the **file_name** attribute.

3. **add_employee Method:**

- This method takes an **Employee** object as an argument and appends its details to the file.
- It opens the file in append mode and writes the employee's information as a comma-separated line.

4. **view_employees Method:**

- This method reads the employee data from the file and prints the details of each employee.

- It opens the file in read mode and iterates through each line. For each line, it splits the data by commas, creates an **Employee** object, and then prints its information.

5. **search_employee Method:**

- This method takes an employee ID as an argument and searches for an employee with that ID in the file.
- It opens the file in read mode, iterates through each line, and checks if the employee ID matches. If found, it returns an **Employee** object with the matching details; otherwise, it returns **None**.

6. **main Function:**

- This is the main execution function that handles the user interface and program flow.
- It creates an instance of **EmployeeManagementSystem** with the file name "employees.txt".
- It enters a loop where the user is presented with options to add employees, view the list of employees, search for an employee, or exit the program.

7. **User Interface:**

- The program displays a menu to the user with options numbered from 1 to 4.
- Depending on the user's choice, the program takes relevant actions:
 - For option 1, it collects employee information, creates an **Employee** object, and adds it to the file using the **add_employee** method.
 - For option 2, it displays the list of employees using the **view_employees** method.
 - For option 3, it searches for an employee using the **search_employee** method and prints their information.
 - For option 4, it exits the program.

8. Execution:

- The `if __name__ == "__main__":` block ensures that the **main** function is executed only when the script is run directly (not when imported as a module).

Course Contents :-

1. Introduction to Python
 - Python Language Features
 - Variables and Assignment
 - Expressions, Statements and Indentation
 - Input and Output
 - Basics of Python Programming
2. Data Types
 - Data Types
 - Numbers and Strings
 - Understanding Relational and Equality operators
 - Lists, Sets and Tuples
 - Dictionaries
 - Type Conversion
3. Operators in Python
 - Arithmetic Operators
 - Comparison Operators and Assignment Operators
 - Bitwise Operators and Logical Operators
 - Membership Operators and Identity Operators
 - Extra problem-solving exercises
4. Algorithmic Problem Solving
 - Building Blocks of Algorithms
 - Pseudo Code and Flow Charts
 - Programming Language, Algorithmic Problem Solving, Simple
 - Strategies for Developing Algorithms
 - Problem Solving Exercises
5. Control Statements
 - if, if-else, if-elif-else statements
 - break, continue and pass statements
 - Problem Solving Exercises

6. Numbers

- Introduction to Numbers
- Number Type Conversion
- Mathematical Functions and Random Number Functions
- Trigonometric Functions, Mathematical Constants

7. Strings

- Introduction to Strings
- String Operations
- Built-in String Methods

8. Lists

- Introduction to Lists
- Basic List Operations
- Built-in List Functions
- List Methods

9. Tuples

- Introduction to Tuples
- Basic Tuple Operations
- Built-in Tuple Functions

10. Dictionaries

- Introduction to Dictionaries and Basic Dictionary Operations
- Built-in Dictionary Functions
- Built-in Dictionary Methods

11. Sets

- Introduction to Sets and Basic Set Operations
- Built-in Set Functions and Methods

12. Comprehensions

- List Comprehensions
- Set Comprehensions
- Dictionary Comprehensions

13. Functions

- Basics of Functions
- Parameters and Arguments
- Keyword Arguments and Default Arguments
- Variable-length arguments
- Anonymous Functions
- Working with multi-dimensional arrays and pointers
- Fruitful Functions
- Global and Local Variables
- Function Composition and Recursion
- Exercises-Functions

14. Modules

- Creation of Modules
- Import Statement
- From import Statement

15. Python Packages

- Introduction to PIP
- Namespace
- Using Python Packages
- Installing Packages via PIP

16. File Handling

- File Path
- File Operations

17. Object Oriented Programming in Python Classes

- Classes
- Self Variable
- Constructors
- Methods
- Inheritance
- Overriding Methods
- Data Hiding

18. Exception Handling

- Errors and Exceptions
- Error Handling
- Handling an Exception
- try-finally Clause
- Raising an Exception
- User-Defined Exceptions

19. Standard Library

- Math and Operating System Interface
- String Pattern Matching
- Internet Access
- Dates and Times
- Multi Threading
- Turtle Graphics
- Data Compression

20. Data Structures

- Introduction
- Arrays
- Stack
- Queue

21. GUI Programming

- Introduction to GUI Programming
- Button, Canvas, Check button, Entry, Frame, Label and List box
- Menu Button, Menu, Message, Radio button, Scale and Scrollbar

LEARNING OUTCOMES

Learning Python offers foundational programming skills, fostering a deep understanding of variables, loops, functions, and object-oriented concepts. Python's readability and versatility empower me to build web apps, analyze data, create AI solutions, and automate tasks. I gain problem-solving abilities, tapping into a supportive community and diverse resources. This knowledge opens doors to web development, data science, machine learning roles, and more, bolstered by Python's cross-platform compatibility and extensive libraries. Ultimately, Python equips you with a valuable skill set for both professional growth and personal projects, making it an ideal choice for aspiring programmers and seasoned developers.

Conclusion:-

In conclusion, embarking on the journey of learning Python is a decision that promises a multitude of benefits and exciting possibilities. As one of the most widely used and versatile programming languages, Python offers a rich learning experience that extends well beyond the realm of coding.

Through Python, you not only gain a solid foundation in programming principles, but you also acquire a language known for its readability and expressiveness. This readability not only accelerates the learning curve but also encourages collaborative coding practices, a crucial aspect in modern software development.

Python's adaptability is its hallmark. From web development to data analysis, machine learning to scientific research, Python has established itself as a go-to language in various domains. Learning Python means acquiring a toolset that enables you to tackle diverse challenges, whether it's crafting elegant web applications or extracting insights from complex datasets.

The vibrant Python community adds another layer of value. Abundant resources, tutorials, and open-source projects are at your disposal, nurturing continuous learning and skill enhancement. This inclusive and collaborative ecosystem ensures that you're never alone on your learning journey.

As you master Python, you cultivate not only technical prowess but also problem-solving skills, analytical thinking, and creativity. These skills transcend programming and can be applied to various facets of life. In a technology-driven world, proficiency in Python translates into enhanced career opportunities, spanning industries and roles.

In essence, learning Python isn't just about acquiring coding proficiency; it's about joining a dynamic community, fostering innovation, and embracing a language that empowers you to turn ideas into reality. Whether you're an aspiring developer, a data enthusiast, or a seasoned programmer, Python opens doors to a world of endless learning and boundless potential.

REFERENCES

- [Python Tutorial | Learn Python Programming \(geeksforgeeks.org\)](https://www.geeksforgeeks.org/python-tutorial/)
- [Learn Python Programming from Scratch - Python Geeks](https://www.pythonforbeginners.com/learn/python-programming-from-scratch/)
- [Introduction to Python \(w3schools.com\)](https://www.w3schools.com/python/)
- [Python Math \(w3schools.com\)](https://www.w3schools.com/python/math.asp)
- [Python Tutorial | Tutorialspoint](https://www.tutorialspoint.com/python/)