./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
| 1. | 13/03/23 | Jai Kishan |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc129592171)

[MODULE -1 [ESSENTIALS OF PYTHON PROGRAMMING] 4](#_Toc129592172)

[Summary 5](#_Toc129592184)

[Challenges faced and how were they overcome 5](#_Toc129592185)

[MODULE -2 [OBJECT ORIENTED PROGRAMMING WITH PYTHON] 6](#_Toc129592187)

[Summary 6](#_Toc129592198)

[Challenges faced and how were they overcome 6](#_Toc129592199)

[MODULE -3 [MANAGING DATABASE OBJECTS IN MYSQL] 4](#_Toc129592172)

[Summary 5](#_Toc129592184)

[Challenges faced and how were they overcome 5](#_Toc129592185)

[MODULE -4 [MASTERING C PROGRAMMING] 4](#_Toc129592172)

[Summary 5](#_Toc129592184)

[Challenges faced and how were they overcome 5](#_Toc129592185)

[MODULE -5 [BASICS OF THREAD AND IPC IN LINUX OS] 4](#_Toc129592172)

[Summary 5](#_Toc129592184)

[Challenges faced and how were they overcome 5](#_Toc129592185)

[MODULE -6 [ESSENTIALS OF OBJECT ORIENTED CONCEPTS USING C++] 4](#_Toc129592172)

[Summary 5](#_Toc129592184)

[Challenges faced and how were they overcome 5](#_Toc129592185)

[Miniproject -1 [TEAM] 6](#_Toc129592187)

[Module/s 6](#_Toc129592188)

[Topic and Subtopics 6](#_Toc129592189)

[Objectives & Requirements 6](#_Toc129592190)

[Design 6](#_Toc129592191)

[Test Plan 6](#_Toc129592192)

[Implementation Summary 6](#_Toc129592193)

[Git Link 6](#_Toc129592194)

[Git Dashboard 6](#_Toc129592195)

[Summary 6](#_Toc129592196)

[Individual Contribution & Highlights 6](#_Toc129592197)

[Summary 6](#_Toc129592198)

[Challenges faced and how were they overcome 6](#_Toc129592199)

**MODULE -1 [Essentials of Python Programming]**

**Summary:**

* We covered the basics of Python, such as its syntax, keywords, and how to install and run Python on your computer.
* We covered the different data types in Python and how to use variables to store and manipulate data.
* We covered the different types of operators in Python and how to use them in code.
* We covered the different types of control flow statements in Python, such as if/else statements and loops, and how to use them to control the flow of your code.
* We covered the different data structures in Python, such as lists, tuples, and dictionaries, and how to use them to store and manipulate data.
* We covered how to define and use functions in Python, including how to pass arguments to functions and how to return values from functions.
* We were given hands-on problems daily to apply the concepts learned in the course.

**MODULE -2 [Object Oriented Programming with Python]**

**Summary:**

* We learned how to use regular expressions in Python using the 're' module.
* We learned about the various features of regular expressions, including character classes, quantifiers, grouping, and capturing.
* We learned about the Search function which is used to search for a specific pattern in each string, and in this course, we learned how to use it effectively.
* We learned about the modifying Strings which is used to modify strings in Python, including concatenation, slicing, and stripping.
* We learned about the Match Methods which is used to search for a pattern at the beginning of a string, and in this course, you learned about the 'match' method.
* We learned about Findall and Finditer functions which are used to find all occurrences of a pattern in a string, and in this course, you learned how to use them.
* We learned about Exception Handling which refers to the process of handling errors and exceptions and how to handle exceptions using the 'try' and 'except' blocks.
* We learned about Pytest which is a testing framework for Python, used to test the functionality of Python code and how to use Pytest to write unit tests for Python code.
* We learned about Pytest Markers which are used to mark certain tests as belonging to a specific category, and in this course, we learned how to use them.
* We learned about Fixtures which are used to set up the environment for your tests.
* We learned about Mock Objects which are used to create fake objects for testing purposes.
* We were given hands-on problems daily to apply the concepts learned in the course.

Overall, this course provided you with a solid foundation in object-oriented programming with Python, including regex, string manipulation, modules and packages, exception handling, and software testing using Pytest.

**MODULE -3 [Managing Database Objects IN MYSQL]**

**Summary:**

* We learned the basics of MySQL, including how to install and set up MySQL, creating and managing databases, tables, and views, and how to insert, update, and delete data from the database.
* We learned the various built-in functions that MySQL provides for data manipulation and processing like string functions, numeric functions, date and time functions, and various other useful functions.
* We learned to use joins and subqueries to retrieve data from multiple tables based on specific criteria. about inner, left, right, and full outer joins, and how to use subqueries to filter data based on complex conditions.
* Then we covered the advanced features of MySQL, including how to create and manage stored procedures and triggers in which we learn how to write SQL code that can be stored in the database and executed later.

Overall, this course would have provided you with a comprehensive understanding of MySQL and how to create and manage databases, tables, views, and stored procedures in MySQL. You would have also learned about advanced topics such as indexing, performance optimization, and triggers, which are important for developing scalable and efficient MySQL applications.

**MODULE -4 [Mastering C Programming]**

**Summary:**

* We learned how to use branching statements like if, else, switch, and loops like for, while, and do-while to control the flow of your program and how to use nested loops and break and continue statements.
* We covered arrays and pointers, how to declare and initialize arrays, access array elements, and use pointers to manipulate data.
* We learned about function prototypes, return values, and parameters, recursion and how to pass arrays and pointers to functions.
* We learn how to declare and initialize strings, access individual characters in a string, and use string manipulation functions.
* We learned how to create our own user-defined data types in C programming, such as structures and unions.
* We covered storage classes in C programming, including auto, static, register, and extern and we learn how to use these storage classes to control the lifetime and visibility of variables.

Overall, this course covers a wide range of topics in C programming, from basic concepts like variables and operators to advanced topics like user-defined data types and preprocessor directives. By the end of this course, we have a solid foundation in C programming and be able to write efficient and effective programs in this language.

**MODULE -5 [Basics of Thread and IPC IN Linux OS]**

**Summary:**

* We learned about basic Linux commands that are frequently used in managing processes, threads, and inter process communication.
* We covered the fundamental concepts of Operating System (OS) and the architecture of the Linux OS and learned about kernel, user space, system calls, process management, and resource allocation.
* We learned about process management, including how to create, terminate, manage processes and learn about signals, which are used to communicate between processes and between the OS and processes.
* We covered the basics of multithreading, including thread creation, synchronization, communication and learned about the pthreads library and how to use it to create and manage threads.
* We were given hands-on exercises and programming assignments to reinforce your understanding of the concepts covered in the course.

Overall, this course provides a comprehensive introduction to the basics of thread and IPC in Linux OS. We learned how to create and manage processes and threads, how to communicate between processes using various IPC techniques, and how to optimize system performance using scheduling policies and memory management techniques.

**MODULE -6 [Essentials of Object-Oriented Concepts using C++]**

**Summary:**

* We learned the basics of functions in C++, including function prototypes, function calls, function parameters, function overloading, and recursion.
* We covered the fundamental concepts of Object-Oriented Programming (OOP) such as encapsulation, inheritance, and polymorphism.
* We covered the concepts of constructors and destructors in C++, including their syntax, use cases, and common pitfalls and learned how to create and use constructors and destructors to initialize and clean up objects in C++ programs.
* We covered the concept of operator overloading in C++, including operator overloading syntax, operator overloading for built-in and user-defined types, and common operator overloading scenarios.
* We covered the concepts of inheritance and polymorphism in C++, including inheritance syntax, base and derived classes, virtual functions, and polymorphic behavior and d how to create and use inheritance and polymorphism to implement OOP concepts and solve problems using OOP techniques.
* We covered the concept of generic programming in C++, including templates, function templates, class templates, and template specialization.
* We covered the basics of file handling in C++, including file input/output, file opening modes, and error handling.

Overall, the course on Object Oriented Concepts using C++ covers a wide range of topics related to OOP and C++ programming, including functions, classes and objects, operator overloading, inheritance and polymorphism, generic programming, file handling, and unit testing.

**Miniproject -1 [Team]**

**Module/s:**

Python

**Topic and Subtopics:**

Cab Booking System:

Booking a cab

Cancelling a booking

Checking fares

Showing bookings

Showing available cabs

Clearing the screen

Menu

About

**Objectives:**

To develop a console application for booking and managing cabs.

To create a MySQL database to store cab details and booking information.

To use object-oriented programming concepts such as classes, objects, inheritance, overloading, and overriding of methods.

To implement regular expressions for searching sources and destinations.

To handle exceptions in all functions.

To create test files with adequate assertions to verify the implementation matches the requirements.

To assess code quality using PyLint.

To submit the project artifact using GitHub.

Design:

**System Level UML:**

Main.py

Checks.py

InsertData.py

User\_Functions.py

Other.py

Subsystem Level UML:

Cab Details table in MySQL database

Test Plan:

**Integration level:** Testing of the entire system after all modules are integrated.

**Unit level:** Testing individual modules to ensure they work correctly.

**Implementation Summary:**

The cab booking system is a console application developed in Python. The application allows users to book a cab, cancel a booking, check fares, show bookings, show available cabs, clear the screen, view the menu, and see information about the application. The system uses a MySQL database to store cab details and booking information. The application is object-oriented, using classes and objects, and implements regular expressions for searching sources and destinations. All functions handle exceptions and have test files with adequate assertions. The code quality is assessed using PyLint.

**GitGEA links:-** <https://gitgea.ltts.com/genesis/system_software/pythonq3fy23/40032355.git>