Course Learning Outcomes

On successful completion of the course, students will be able to:

1. solve simple programming problems using iteration and selection, and basic constructs: structures, arrays and functions.
2. create classes and their objects and use access specifiers for data hiding depicting advantage of Abstraction.
3. construct classes for code reusability depicting advantage of Inheritance.
4. implement Function Overloading depicting advantage of Polymorphism.
5. create file, read/write from/to files.

Unit 1

Introduction to C++: Need and characteristics of Object-Oriented Programming, Structure of a C++ Program (main () function, header files, output, input, comments), compile and execute a simple program.

Unit 2

Data types and Expression: Keywords, built in data types, variables and constants, naming convention, Input-Output statements, expressions and operators, precedence of operators, typecasting, library functions.

Unit 3

Control Constructs in C++ : Decision making using selection constructs, looping constructs , control constructs.

Unit 4

User defined Data types and functions: User defined data types, defining and initializing structures, derived data types, defining and initializing single and multi dimensional arrays, and user defined functions, passing arguments to functions, returning values from functions, inline functions, default arguments.

Unit 5

Classes and Objects: Need of abstraction, encapsulation, inheritance and polymorphism, creating classes, objects as function arguments, modifiers and access control, constructors and destructors, Implementation of single level inheritance, implementation of polymorphism, function overloading.

Unit 6

File Handling: File I/O Basics, read and write operations.

Practical

1. Write a program to find the largest of n natural numbers.
2. Write a program to find whether a given number is prime or not.
3. Write a menu driven program for following:
   a) display a Fibonacci series
   b) compute Factorial of a number
   c) to check whether a given number is odd or even.
   d) to check whether a given string is palindrome or not.
4. Write a program to print the sum and product of digits of an Integer and reverse the Integer.
5. Write a program to create an array of 10 integers. Accept values from the user in that array. Input another number from the user and find out how many numbers are equal to the number passed, how many are greater and how many are less than the number passed.
6. Write a program that will prompt the user for a list of 5 prices. Compute the average of the prices and find out all the prices that are higher than the calculated average.
7. Design a class named Car, having registration number, model and engine as its private members. Here engine is an object of a class called Engine with the private members: Chassis number and make. Define a suitable constructor of Car and override toString() Method to print the details of a car. Assume appropriate data types for the instance Members of the classes. Write a Java program to test the above class.
8. Write a program that computes the area of a circle, rectangle and a Cylinder using function overloading.

References

1. Lafore, R. *Object Oriented Programming in C++ (4th Edition).* SAMS Publishing.

Additional Resources

1. Balaguruswamy, E. (2017). *Object Oriented Programming with C++ (7th edition).* McGraw-Hill Education.
2. Kanetkar, Y. P. (2015). *Let us C++ (2nd Edition).* BPB Publishers.
3. Stroustrup, B. (2013). *The C++ Programming Language (4th Edition).* Pearson Education.

Teaching Learning Process

- Talk and chalk method
- Computer based presentations by teachers.
- Group Discussions
- Assignments
- Offline and online Quiz
- Presentations by group of students for enhanced learning.

Tentative weekly teaching plan is as follows: