

ITEX 103 Object Oriented Programming

Course Title: **Object Oriented Programming**

Course Code: **ITEX 103**

Semester: II

Credit Hours: 3 (2+1)

Nature of the course: T/P

1. Course Description and Goals

The course covers the concepts of Object-Oriented Programming using C++ programming language. The main objective of this course is to deliver and to make learners understand the basic concepts of OOP, release the peculiarity of Object-Oriented Paradigm and implement these concepts in real world projects. It covers the OOP concepts like objects, classes, polymorphism, overloading, inheritance, exceptions, file handling and more.

2. Learning Outcomes

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

- i) understand the specialty and importance of Object-Oriented Approach
- j) implement the programming features of OOP
- k) create programs and projects using C++ programming language
- l) learn any other OOP language based upon the basics delivered through this course

3. Content with specific objectives

Specific objectives	Contents
<ul style="list-style-type: none">1. Compare procedure and object oriented programming concept2. Describe the feature of object-oriented programming.3. List different types of OOPs4. Compare coding structure of C and C++.5. Understand the details of C++ program structure.6. Revise the concepts of control structures in C++	Unit One: Introduction to OOP (5 Hrs) <ul style="list-style-type: none">8. Software Crisis and emergence of OOP9. Procedural vs OOP Languages10. Features of OOPs11. Programming structures in C++12. Basic programming terminologies: tokens, data types, operators etc.13. Programming Construct<ul style="list-style-type: none">Control Structures
<ul style="list-style-type: none">1. understand the basics of classes and objects2. be able to initialize objects in accordance to its class	Unit Two: Object and Classes (9 hrs.) <ul style="list-style-type: none">21. Introduction to Objects and Classes22. Defining a class with data members and member functions

<ul style="list-style-type: none"> 3. implement the use of this operator 4. understand the idea of access specifiers 5. understand and implement the use of constructor and destructors in C++ 	<ul style="list-style-type: none"> 23. Access specifiers 24. Initializing objects 25. Static Data Members 26. Static Member Functions 27. this operator 28. Constructor 29. Destructor
<ul style="list-style-type: none"> 1. To understand and demonstrate the use of operator overloading in C++ 2. implement unary operator overloading 3. implement binary operator overloading 4. implement the concept of operator overloading in member functions and non-member functions 	Unit Three: Operator Overloading (7 hrs.) <ul style="list-style-type: none"> 1. Introduction to operator overloading 2. Overloadable operators 3. Syntax and rules of operator overloading 4. Unary operator overloading 5. Binary Operator overloading 6. Operator overloading with Member and non-member functions 7. Overloading binary operator using a friend function
<ul style="list-style-type: none"> 1. understand and implement the concept of inheritance 2. understand base class and derived classes 3. implement constructors and destructors for derived class in C++ 4. understand concept of abstract class and virtual base class 5. implement use of access specifiers in C++ programs 	Unit Four: Inheritance (8 hrs.) <ul style="list-style-type: none"> 1. Introduction to inheritance 2. Base class and derived class 3. Derived class constructor 4. Derived class destructor 5. Single inheritance and multiple inheritance 6. Virtual Base Classes and Abstract Classes 7. Public, Private and Protected access and control
<ul style="list-style-type: none"> 1. Understand the concept of polymorphism 2. implement new and delete operators 3. understand and implement virtual functions in C++ programs 	Unit Five: Polymorphism (9 hrs.) <ul style="list-style-type: none"> 1. Introduction to polymorphism 2. New and Delete operator 3. Pointer to objects 4. Pointer to derived class 5. Virtual Functions 6. Friend Functions

4. understand the concept of dynamic binding	7. dynamic cast operator 8. typed operator
1. Understand the idea of templates and frameworks 2. to implement class templates in C++ 3. understand function templates	Unit Six: Templates (5 hrs.) 1. Concept of templates 2. Class templates 3. Function Templates 4. Concept of Frameworks
1. Understand the idea of error and exceptions 2. implement ideas to handle exceptions in C++ 3. implement rethrowing an exception in C++ 4. handle various types of exceptions in C++	Unit seven: Exception Handling (5 hrs.) 1. Error Handling 2. Exception handling constructs (try, catch, throw) 3. Multiple exception handling 4. Catching all exceptions 5. Exception with arguments 6. Re-throwing an exception 7. Handling uncaught and unexpected exceptions

8. Major Teaching and Learning Strategy

This course is designed to be implemented in both practical and theoretical basis.

Understanding the basics under facilitators guidance and supervision by being involved in classroom activities will be used to complete the theoretical requirements.

Implementation of the theoretical knowledge in the computer room or in personal computers will entertain the practical requirements of the course. Both the learner and the facilitator are expected to progress in developing depth knowledge regarding the subject matter in time to come.

9. Assessment Plan

a. In-semester:

Students' evaluation will be carried out throughout the semester according to their performance and participation in classroom activities related to the course. The classroom activities may include two or more of any of the following:

1. presentations
2. project works
3. practical
4. group discussions
5. knowledge sharing
6. internal examinations/tests etc.

b. End-semester

10. References

- c. Deitel and Deitel, C++ How to Program, Latest Edition, Pearson Publication.
- d. Joyce Farrell, Object-oriented programming using C++, Fourth Edition, Cengage Learning
- e. Daya Sagar Baral and Diwakar Baral, “The Secrets of Object-Oriented Programming in C++”, Latest Edition.