

ACADEMIC PLAN FOR IV SEMESTER

CODE NO.: ETCS 210

PAPER: OBJECT ORIENTED PROGRAMMING

L	T	C
3	0	3

	Topics	No. of Lectures (36)
1	Introduction: Introducing Object-Oriented Approach related to other paradigms (functional, data decomposition)	1
2	Characteristics of Object-Oriented Languages	1
3	Basic terms and ideas: Abstraction, Encapsulation, Information hiding, Inheritance, Polymorphism	1
4	Review of C, Difference between C and C++, cin, cout, new, delete operators	2
5	Classes and Objects: Abstract data types, Object & classes, attributes, methods	1
6	C++ class declaration, State identity and behavior of an object	2
7	Constructors and destructors	1
8	instantiation of objects, Default parameter value, Copy Constructor	2
9	Static Class Data, Constant and Classes	1
10	C++ garbage collection, dynamic memory allocation	2
End First Term		
11	Inheritance and Polymorphism: Inheritance	1
12	Types of Inheritance, Class hierarchy	1
13	derivation – public, private & protected	1
14	Agrégation, composition vs classification hiérarchies,	1
15	Polymorphism, Type of Polymorphism – Compile time and runtime,	1
16	Method polymorphism, Polymorphism by parameter	1
17	Operator overloading	1
18	Parametric polymorphism	1
19	Generic function – template function	1
20	Function name overloading,	1
21	Overriding inheritance methods	1
22	Files and Exception Handling: Persistent objects	1
23	Streams and files, Namespaces	1
24	Exception handling, Generic Classes	2
25	Standard Template Library: Standard Template Library	1
26	Overview of Standard Template Library, Containers,	1
27	Algorithms, Iterators	2
28	Other STL Elements, The Container Classes,	2
29	General Theory of Operation, Vectors	1

**AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
BIJWASAN, NEW DELHI-61**

Code No.: ETCS 210

Paper: Object Oriented Programming

L	T	C
3	0	3

INSTRUCTIONS TO PAPER SETTERS:

MAXIMUM

MARKS: 75

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 25 marks.
2. Apart from question no. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be of 12.5 marks

UNIT – I:

Objects, relating to other paradigms (functional, data decomposition), basic terms and ideas (abstraction, encapsulation, inheritance, polymorphism). Review of C, difference between C and C++, cin, cout, new, delete operators. [T1,T2][No. of hrs. 11]

UNIT – II:

Encapsulation, information hiding, abstract data types, object & classes, attributes, methods. C++ class declaration, state identity and behavior of an object, constructors and destructors, instantiation of objects, default parameter value, object types, C++ garbage collection, dynamic memory allocation, meta class/abstract classes. [T1,T2][No. of hrs. 11]

UNIT – III:

Inheritance, Class hierarchy, derivation – public, private & protected; aggregation, composition vs classification hierarchies, polymorphism, categorization of polymorphic techniques, method polymorphism, polymorphism by parameter, operator overloading, parametric polymorphism, generic function – template function, function name overloading, overriding inheritance methods, run time polymorphism. [T1,T2][No. of hrs. 11]

UNIT – IV:

Standard C++ classes, using multiple inheritance, persistent objects, streams and files, namespaces, exception handling, generic classes, standard template library: Library organization and containers, standard containers, algorithm and Function objects, iterators and allocators, strings, streams, manipulators, user defined manipulators, vectors, valarray, slice, generalized numeric algorithm. [T1,T2][No. of hrs. 11]

Text Books:

[T1] Rumbaugh et. al. "Object Oriented Modelling & Design", Prentice Hall

[T2] A.R.Venugopal, Rajkumar, T. Ravishanker "Mastering C++", TMH

Reference Books:

[R1] A.K. Sharma, "Object Oriented Programming using C++", Pearson

[R2] G . Booch "Object Oriented Design & Applications", Benjamin,Cummings.

[R3] E.Balaguruswamy, "Objected Oriented Programming with C++", TMH

[R4] S. B. Lippman & J. Lajoie, "C++ Primer", 3rd Edition, Addison Wesley, 2000.

[R4] R. Lafore, "Object Oriented Programming using C++", Galgotia.

[R5] D . Parsons, "Object Oriented Programming with C++",BPB Publication.

[R6] Steven C. Lawlor, "The Art of Programming Computer Science with C++", Vikas Publication.

TUTORIAL – 1

- 1) Explain the following:
 - a) Procedure orientation c) object orientation
 - b) Logic orientation d) constraint orientation
- 2) What are the drawbacks of structural programming?
- 3) Write a function string – cat in a program that receive two character strings as argument and concatenates these two strings, the function should return concatenated string.
- 4) Write a program in which, declare pointers to strings representing the days of week, sort string in alphabetical order (sort the pointers to strings and not the actual string)
- 5) Discuss the 3 advantages of oops.
- 6) What is the functionality of Pre-processor directive `#include<iostream>` ?
- 7) What is the difference between OOP and Procedure oriented programming system?
- 8) How do the following statements differ:
 - a) `char* const p;`
 - b) `char const* p;`
- 9) What do you understand by identity of an object? Explain with example.
- 10) Explain the following:
 - a) Association b) Aggregation c) Instantiation d) Inheritance
 - e) using f) Meta Class

TUTORIAL – 2

- 1) What are abstract data types?
- 2) What are mutable variables?
- 3) What do you mean by Inheritance & Polymorphism in C++?
- 4) What is the difference between C & C++.
- 5) Define the terms `printf` & `cout`, `scanf` & `cin`, `malloc` & `new`, `free` & `delete` with comparisons.
- 6) How many data types are available in C++, Explain them with there storage classification?
- 7) How the information is hided in OOPs?
- 8) Which class is `cout` and `cin` an object of?
- 9) What do you understand by modeling? How is object model different from functional model?
- 10) Compare the significance of attributes with operations and methods.

TUTORIAL – 3

- 1) Write a program to generate following output.

ITEMS	COST	VALUE
10	*5	+5.00e+01
*100	*400	+4.00e+04
***5	*100	+5.00e+02

- 2) How does a constant defined by **const** differ from the constant defined by the pre-processor statement **#define**?
- 3) Write a program which accepts an integer from user and uses pointer to a pointer, which points to pointer to int (**p) to display that value.
- 4) Write a program containing structure client having account _no, name, amount as data members. Use pointers to assign value to these variables in function main().
- 5) Define the following:
- this pointer.
 - Interface class.
 - Container class
- 6) Compare link with association. Explain aggregation, generalization and specialization.
- 7) Explain the difference between the events and states based on conditions.
- 8) Explain the classical categorization, conceptual clustering and prototyping in reference to objects.

TUTORIAL – 4

- 1) Why do we use void pointers? Give an example.
- 2) Define a class person. Declare male and female as pointer to person, assign name and age using these pointers and then display results. Use function assign() and display() in person.
- 3) Differentiate between the term Classes & Objects with example.
- 4) What do you mean by attributes & methods. Give an example of method.
- 5) What is this pointer? Where and why the compiler insert it implicitly.
- 6) What is void pointer & wild pointer? Explain with example.
- 7) What are smart pointers? How they are created.
- 8) Find the values of m and n after the following 2 statements are executed:
- ```
int m = 5;
int n = m++ * ++m ;
```

## TUTORIAL – 5

- 1) What is a constructor? How is it different from normal member function?
- 2) Can we have more than one constructor in a class? If yes, how many and explain the need for such a situation.
- 3) What do you mean by dynamic initialization of objects? Why do we need to do this?
- 4) What do you mean by a destructor? Why we need this?
- 5) Explain why static data members should be explicitly declared outside the class.
- 6) What is a nested class? What is its use?
- 7) Why should the formal argument of a copy constructor be a reference object?

## TUTORIAL – 6

- 1) What do you mean by inheritance in C++? What are the different forms of inheritance? Give an example.
- 2) What do you mean by early binding and late binding. Explain advantages and disadvantages of each.
- 3) Class D is derived from class B. the class D does not contain any data members of its own. Does the class D requires constructors? If yes, explain why.
- 4) Write a program to define three classes A, B, C. Each class contains private data members. Derive class C from A and B by applying multiple inheritance. Read and display the data using constructors and destructors.
- 5) What is dynamic memory allocation? How is it different from static memory allocation.
- 6) Distinguish between the following 2 statements:-  
time T2(T1);  
time T2 = T1;                      T1 and T2 are objects of **time** class.

## TUTORIAL – 7

- 1) What does polymorphism mean in C++? How is polymorphism achieved at a) compile time b) run time? Explain with example.
- 2) What does **this** pointer point to? What are the applications of **this** pointer?
- 3) What is a pure virtual function? How is it different from normal virtual function? What is the use of pure virtual function?
- 4) Write a program, to find the number of zero elements in 3x3 matrix. Read and the elements in matrix form. Use pointers only.
- 5) In which order are the constructor's and destructors called when an object of the derived class is created.
- 6) What is a virtual base class? When do we make a class virtual? What is an abstract class?
- 7) Compare the following library function:
  - 1) strcmp() and strncmp()
  - 2) strcat() and strncat()
- 8) Write a program to calculate the area of rectangle and circle using run time polymorphism.
- 9) Explain the following:
  - a) Parameterized classes      c) Meta classes
  - b) Nesting classes              d) Class utilities

## TUTORIAL – 8

- 1) What do you mean by operator overloading? What are the rules for overloading operator?
- 2) How are friend functions used to carry out overloading of operators? When is a friend function compulsory in operator overloading? Give an example.
- 3) A friend function cannot be used to overload the assignment operator =. Explain why?
- 4) Write a program to convert radian into degree and vice versa using conversion functions.
- 5) What is the difference between text file and binary file?
- 6) Why to use a friend class?
- 7) Create a student class and implement operator overloading.
- 8) Explain the classification of polymorphic technique using diagram.

## **TUTORIAL – 9**

1. What is generic programming? What are its advantages and applications?
2. What is a class template and a function template? Write a function template for bubble sort (not inline)?
3. Create a template for bubble sort function.
4. What do you mean by overriding and overloading of functions in template and inheritance? Explain with example.
5. Write a program using find algorithm to locate the position of a specified value in a sequence container.
6. What are smart pointers? How are they created?
7. How can a class object be converted to a value of fundamental data types?
8. Write a program to add, delete and print item in a vector.
9. Merge 2 arrays using vectors.

## **TUTORIAL – 10**

1. What is the syntax for indicating a list of exceptions that a function can raise? What happens if an unspecified exception is raised?
2. What are the limitations of exception handling in C++?
3. What do mean by re-throwing of an exception? What is the role of try, catch and throw keywords in exception handling?
4. Write a program to explain the concept of multiple catch statements. Also use re-throw and catch all.
5. Write a program for addition of numbers passed through command line.
6. What is a file? Which are the different types of file opening modes? List their names with meaning.
7. What are the streams? Explain the features of C++ I/O stream.
8. Write an interactive program that accepts students Name, class and score and prints the result to a file.
9. What if the program does not provide a catch clause for the exception that is thrown?
10. Write a program to demonstrate rethrowing an exception.