

OBJECT ORIENTED PROGRAMMING AND DESIGN

QUESTION BANK

UNIT – I

OBJECT AND CLASSES

PART – A (2 MARKS)

1. Define Object Oriented Programming.
2. Define Objects.
3. What are the features of Object oriented programming.
4. Define Encapsulation and Data hiding.
5. Define Data Abstraction.
6. Define Data members.
7. Define Member functions.
8. Define Inheritance.
9. Define Polymorphism.
10. List and define the two types of Polymorphism.
11. Define Dynamic Binding.
12. Define Message Passing.
13. List some benefits of OOPS.
14. List out the applications of OOP.
15. What is the return type of main ()?
16. List out the four basic sections in a typical C++ program.
17. Define Token. What are the tokens used in C++?
18. Define identifier. What are the rules to be followed for identifiers?
19. State the use of void in C++.
20. Define an Enumeration data type.

21. Define reference variable. Give its syntax.
22. List out the new operators introduced in c++.
23. What is the use of Scope resolution operator?
24. List out the memory referencing operators.
25. Define Implicit Conversion.
26. What is call by reference?
27. What are inline functions?
28. State the advantages of Default Arguments.
29. Define Function overloading.
30. Define friend function.
31. Write the limitations/ disadvantages of C++
32. What is meant by Lvalue and Rvalue ?

PART – B

1. Explain the basic concepts of Object oriented programming (11)
2. Explain the use of constant pointers and pointers to constant with an example.(11)
3. a. State the differences between class and struct and also illustrate with an example. (6)
b. What are the difference between pointers to constants and constant to pointers? (6)
4. a. Write a C++ program using inline function. (6)
b. Write a C++ program to illustrate the static function (6)
5. Explain briefly about function overloading with a suitable example. (11)
6. a. Discuss constant and volatile functions. (6)
b. What is linkage specification and also explain its need. (6)
7. Explain about call by reference and return by reference with program. (11)

8. Explain Nested classes and local classes with an example (11)
9. Explain in detail about control structures with eg.(11)
10. Write a program to evaluate the following function

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$
- 11 a) What is a friend function . What are the merits and demerits of using friend function?(5)
 b) Write a program to explain the inline function (6)
12. a) Explain the structure of C++ program (5)
 b) Write a program to explain the concept of array of objects (6)
13. Explain in detail about formatted and unformatted console I/O operations (11)
14. a) Write about inline function (5).
 b) Discuss about access specifier. Write about declaring member function inside and outside a class (6)

UNIT – II CONSTRUCTORS AND OVERLOADING

PART – A (2 MARKS)

1. Define Constructor.
2. List some of the special characteristics of constructor.
3. Give the various types of constructors.
4. What are the ways in which a constructor can be called?
5. What is meant by dynamic initialization of objects.
6. Define Destructor.
7. List some of the rules for operator overloading.
8. What are the types of type conversions?
9. What are the conditions should a casting operator satisfy?
10. How the objects are initialized dynamically?
11. Define abstract class.
12. Define virtual base class

PART – B

1. a. Explain the copy constructors with an example? (6)
- b. Explain explicit Constructors, Parametrized Constructors, and multiple Constructors with suitable example. (6)
2. a. How to achieve operator overloading through friend Function? (6)
- b. Write a program using friend functions for overloading <<and>>operators? (6)
3. Explain the wrapper classes with examples? (11)
4. Write a program to add two complex numbers using operator overloading concept
5. Create a class vector to hold an array of integers and perform the following operations using constructor for dynamic allocation of array based on size given as argument
 - a) create two vectors using constructor
 - b) Add the vectors and return the resultant vector
 - c) Multiply all elements in the vector by a given number X and display the resultant vector
6. Write a C++ program to find the area of various 2D shapes such as square, rectangle, triangle, circle and ellipse using function overloading.

INHERITANCE AND POLYMORPHISM

PART – A (2 MARKS)

1. What are types of inheritance?
2. Give the syntax for inheritance.
3. Define single inheritance.
4. Define multi-level inheritance.
5. Define multiple inheritance.
6. What is an abstract class?

PART – B

1. Explain the different types of polymorphism (11)
2. Explain various types of Inheritance. (11)

3. Describe Pure Virtual function with an example. (11)
4. Write a C++ program using this pointer. (11)
5. Write a C++ program using dynamic_const. (11)
6. Explain in detail about cross casting and down casting. (11)
- 7.a) Explain in detail about the various run time casting in C++ (6)
 - b) Write a C++ program for calculating the are of rectangle and circle using run time polymorphism (5)
- 8.a) Write a program in C++ to read two strings and perform the following string manipulation function (7)
 - (i) Find the long string
 - (ii) Compare the two strings
 - (iii) Concatenate them
- b) Explain in detail about dynamic objects. How are they created ? (4)

UNIT – III

Pointers and arrays , Memory , Polymorphism

1. What is a pointer . Give eg.
2. Give the use of new and delete .
3. Write the use of functions malloc() and free()
4. Explain how to create dynamic objects
5. What is static binding?
6. What is dynamic binding ?
7. What are the types of polymorphism?
8. Define 'this' pointer.
9. What is virtual function?
10. When a function is declared as virtual, C++ determines which function to
11. What is pure virtual function?
12. Define RTTI.
13. What is cross casting?

14. What is down casting?
15. How is a string object created ?

UNIT – IV

FILE MANIPULATION

PART – A (2 MARKS)

1. Define manipulators and also mention the manipulators that are used in C++.
2. What is the need for streams?
3. List some predefined streams.
4. What are the possible types that a file can be defined?
5. What are the two methods available for opening the files?
6. What is global namespace?
7. Write any four operations possible on string objects.
8. What are the advantages of using generic algorithm?
9. What is STL?
10. What is find()?
11. What are the ways that a string object can be created?

PART – B

1. Explain about Formatted and Unformatted IO with suitable Example (11)
2. What is manipulator? Difference between manipulators and ios Function? (11)
3. Explain the process of open,read,write and close files? (11)
4. Explain the role of seekg(),seekp(),tellg(),tellp(),function in the process of random access in a binary file (11)

5. Explain about the STD Namespace (11)
6. Explain the Standard Template Library and how it is working? (11)
7. a) Explain in detail about exception handling in C++ with suitable code (6)
b) Write a C++ program using function template to find the product of two integer or floating point type of data(5)
8. Create a class Player with data members are playerid, playername, playergame and member functions are getplayer(), searchplayer(), and displayplayer() . Use file to store N players information and perform those operations
9. Explain the file pointers and file manipulators (11)
10. Write a program to read and write an array of integers into binary file (11)

TEMPLATE AND EXCEPTION HANDLING

PART – A (2 MARKS)

1. What is the need for template function in C++? What are their advantages?
2. What are the drawbacks of using macros?
3. What is Function Template?
4. What are the components of Exception Handling?
5. What is Uncaught Exception?
6. What is the terminate () functions?
7. What are the disadvantages of the exception handling?

PART – B

1. Explain the Function template (11)
2. Explain the class template (11)
3. What is the need for exception handling (11)

4. Explain the following function
5. a) What are specifications? In which case are they needed? (6)
b) What are the disadvantages of the exception handling mechanism? (6)
6. When do we need multiple catch blocks for a single try block? Give an example? (11)
- 7.. Explain function template with good example program(11)
8. Discuss exception handling in C++ (11)

UNIT –V

OBJECT MODELLING AND OBJECT ORIENTED SOFTWARE DEVELOPMENT

PART – A (2 MARKS)

1. What is meant by activity diagram ?
2. Define interaction modelling.
3. What do you know about use case model ?
4. Write any two goodness criteria of object oriented design.
5. What are the advantages of UML ?
6. What are the patterns ?
7. What is the significance of OOPD?
8. Explain swim lane and life lane in UML.
9. State the usage of adapter pattern design.
10. What is the use of use case diagram?
11. What is meant by sequential diagram?
12. What is a state diagram?
13. Explain class diagram with eg.
14. Explain activity diagram.
15. Mention the use of collaboration diagram.
16. What is the use of UML diagrams?

Part – B

1. Draw and explain the class diagram for a simple banking application
2. Explain interaction diagrams with an eg.
3. Derive the use case model for student course registration system in a University with proper interaction and activity diagram which include the following tasks publication

of courses offered, student enrolment for courses, attendance maintenance, conduct of exams and marks maintenance and result publication

4. a) Explain in detail about patterns with an ex. (4)
5. How an activity and state chart diagrams are used for modelling ?(7) Explain the interaction and state chart diagram of UML
6. Explain the concept of Interaction modelling
7. Write a detailed note on use case diagram with suitable eg.
8. Construct an UML use case diagram to withdraw amount in ATM banking.
9. Design sequence and collaboration diagram for issue and return of books in library.