

**CS 6456 OBJECT ORIENTED PROGRAMMING**

**II YEAR EEE**

**Mr.N.NAGULANAND**

**QUESTION BANK**

ஊக்கமது கைவிடேல்

## **UNIT-1 PART A**

1. Lists some characteristics of object-oriented language.
2. With respect to C++ distinguish objects and classes.
3. Evaluate Procedural programming with OO programming.
4. Analyze how data encapsulation supports reusability
5. Create a friend function which violates encapsulation?
6. Classify the native types in C++.
7. Create an example for Enumerated data types.
8. Discuss a statement/Expression with an example?
9. Examine function prototype and give an example.
10. Summarize pointer and outline pointer arithmetic.
11. Compare pointer and a reference?
12. Show the syntax to dynamically allocate memory to a constructor.
13. What is function overloading? List out its advantages.
14. What are inline functions? Give an example.
15. What is an Abstract Data Type? Give an example and how it differs from class?
16. Differentiate between stack and queue and give its applications.
17. How Struct relates with Union/Class. Give an example.
18. Point out the operations supported by the list ADT?
19. Define Type casting.
20. List out various formats of bit field.

## **PART B**

1. How would you declare function to be a constant in C++? What are the properties of such Function? Explain with a demo program.
2. What are inline functions? What are their advantages? Give an example .What is the rules to be followed while defining inline functions?
3. . Examine a program to check how many instances of a class are created using the static Member function.
4. Summarize various type conversions? Explain each with a program.
- 5.. How would you declare function with default argument in C++? Demonstrate with a Demo program.
6. Write an object oriented code where a member function of a class reads a set of numbers up to 'n' and print the contents of the array in reverse order. This function is called by

## UNIT-2

### PART A

1. List the two types of polymorphism? Give an example for each.
2. Recall the operators in C++ which you cannot be overloaded.
3. Classify the use of keywords “void”/”new”/”delete”.
4. Give the use of scope resolution operator.
5. Define an abstract class with an example.
6. Create a copy constructor for class date (assume dd,mm,yy as its members).
7. How are virtual functions declared in C++?
8. Compose pure virtual functions? Where are they used?
9. When will the destructors be called? Is it implicit or explicit?
10. What is a constructor? Is it mandatory to use constructors in a class?
11. Define friend/static/const function with its rules.
12. What is an operator function? Give an example.
13. What is dynamic binding?
14. Classify predefined C++ stream objects?
15. Demonstrate a stream class which is required to create an output stream?
16. Explain the various file stream classes needed for file manipulation.
17. Sketch and Show the I/O stream hierarchy?
18. Why is it necessary to include the file iostream in all our programs?
19. Name the two ways in which a file can be opened.
20. Give the use of “this” Pointer.

## PART B

1. Write a C++ program to perform 2D matrix operations as follows:
  - i) Define class MATRIX, use appropriate constructor(s).
  - ii) Define methods for the following two matrix operations:
    - Determinant and transpose.
  - iii) Write a main program to discuss the use of the constructor in
2. Write a C++ program as follows to perform arithmetic operations on Rational numbers of type a/b, where a and b are integers.
  - (i) Define a class by 'Rational Number'.
  - (ii) Use operator overloaded methods for addition and subtraction.
  - (iii) Write a main program to demonstrate the use of this class and its methods.
  - (iv) Give a sample output.Write a C++ program for the following:
  - (i) Ramesh's basic salary is given input to the keyboard. His dearness allowance is 40% of basic salary and house rent allowance is 20% of basic salary. Write a program to calculate his gross salary. Use constructors, Destructors and member functions.
4. Construct and analyze a class by name 'Box' with a constructor method and volume method. Constructor initialized the length, breath and height of the box objects. Volume method computes the volume of the box using the formula length \* breath \* height. Create three box objects and compute their volume by declaring a pointer to the box (Any Program).
5. Describe the rules of operator overloading with one example and Write a C++ program to define overloaded constructor to perform string initialization, string copy and string destruction.
6. Write an object oriented program in C++ that print a factorial/fibonacci series of a given number using a copy constructor and a destructor member function.
7. Write and explain a C++ program to list out a prime/Armstrong numbers between the two given limits.
8. Develop a function that computes and displays the area of a rectangle and the area of a square. Declare this function as a friend function in Rectangle and square classes and compute the area of a rectangle and square objects from the main function.(Any Program)
9. Consider the TNEB Electricity bill generation problem. Define suitable classes and objects. Write a program to implement the application with near output format.
10. With suitable example, examine how function overloading and operator overloading supports compile-time polymorphism .



## UNIT-3

### PART A

1. Construct the syntax of try – catch- throw in exception handling in C++.
2. Differentiate function template and class template
3. What are the manipulators available in C++?
4. What is a hierarchy inheritance? Give an example.
5. Give any five functions used for file processing.
6. What is the use of templates?
7. What are manipulators? How do you create a one?
8. Give any two examples of exceptions.
9. Why is it not possible to refer an exception declared within try block outside try block?
10. Illustrate the exception handling mechanism.
11. List out any four containers supported by Standard Template Library?
12. What is rethrowing an expression?
13. What is an exception? Give any four examples of exception.
14. Examine virtual base class with an example.
15. Point out the reason templates used in the C++.
16. List the way of class template inheritance.
17. List the different types of inheritance.
18. Summarize the classes in the “iostream.h”.
19. Draw console stream class hierarchy.
20. Explain file pointer and give its types.

### PART B

1. What is a template function? Design and implement a template version functions min() and max() for finding minimum and maximum values of a given set of elements in list T represented by a seqList <T> object.
2. What is the significance of inheritance and explain the various types of inheritance with a programming example for each type.
3. Analyze the exception handling mechanism available in C++ with suitable examples.
4. Write a program that demonstrates several exception types being caught with the catch (...) exception handler.
5. Write an object oriented program in C++ using a class template to read any five parameterized data type such as float and integer, and print the average.
6. Classficy the components of Standard Template Library in detail.
7. Explain the 4 functions Seekg, Seekp, tellg, tellp used for setting pointers during file operation and show how they are derived from fstream class.
8. Describe about generic program in details.

## UNIT-4 PART A

1. What is byte code? Mention its advantage.
2. Give two examples for access modifier.
3. Design a sample statement for `parseInt( )` and give comments for the statement.
4. What is an abstract class?
5. Define a wrapper class in java?
6. Why is java language called as robust?
7. How does java make an executable file?
8. Differentiate Interface and Abstract class?
9. What is an inner class?
10. Examine the significance of Java Virtual Machine?
11. How is garbage collection done in java?
12. What is a token? List the various types of tokens supported by Java.
13. In Java, can a class be defined inside a method? If so what rule governs access to the variables of the enclosing method? If not, give reasons.
14. How does Java achieve platform independence?
15. Distinguish between method overriding and method overloading in Java.
16. Criticize the type of class for which objects cannot be created?
17. Compose the two ways of using Super/this Keyword?
18. How do you use final keyword in variable/function/class in java?
19. How do you define an array with example?
20. Define Java Virtual Machine (JVM).

## PART B

1. Write a java program to create two single dimensional arrays, initialize them and add them; store the result in another array.
2. Write a menu-based java program that can calculate the area of a triangle, circle or square, based on the user's choice.
3. Describe the structure of a typical Java program and give the steps to execute it.
4. Write a java class called '**student**' with name, Marks of 3 subjects and total Marks. Write another class named '**calculate**' that gets the Marks of the student and calculates the total Marks and displays the result (pass or fail).
5. Describe the three different types of inheritance with an example java program for each.
6. Write a java program to find the maximum number of the given array.
7. Write a java to calculate and convert an Integer array to string.
8. Write a simple java program to classify basic Calculator operations.
9. Write a java Program to evaluate the sum of the following series

$$1-2+3-4+\dots+n$$

## UNIT-5 PART A

1. Define interface. State its use.
2. What is thread? How does it differ from a process?
3. Under which context would you use final and finally?
4. Give one example on streams.
5. List the advantage of using packages in Java Programming?
6. What are the similarities between interfaces and classes in Java?
7. How do we set priorities for threads?
8. What happens if an exception handler is not defined when exception is thrown?
9. Explain the life cycle of thread.
10. What are the two methods by which we may stop threads?
11. What is an exception? Give any four examples of exception.
12. What is multithreading?
13. State threads Synchronization.
14. Draw I/O Stream Hierarchy.
15. Compare String and String Buffer.
16. List the various string methods.
17. Express the various ways to import packages.
18. Which method can be used for changing case of characters?
19. Sketch the Exception Hierarchy.
20. How to create a user defined exception with example.

## PART B

1. Write a java program to perform all string operations using the String Class.
2. Demonstrate exception handling in java with an example program.
3. Summarize the following with examples from Java. (i) Streams and IO (ii) Java threads.
4. What is a thread? How do you create threads and explain?
5. Explain the concept of streams and its byte stream classes in detail.
6. How packages are used to resolve naming conflicts in java? With an example show to add classes to packages and how to import packages in classes.
7. What are interfaces? Explain with an example how multiple inheritances are implemented using interfaces.