

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CS6301 - PROGRAMMING AND DATA STRUCTURES II

QUESTION BANK

UNIT - I

1. Write down the characteristics of object oriented programming.

Hints:

- Encapsulation
- Data abstraction
- Inheritance
- Polymorphism
- Message passing
- Extendibility

2. Explain the elements of object oriented programming.

Hints:

- Classes
- Objects
- Constructor
- Destructor
- Dynamic binding

3. Describe the applications of OOP technology.

Hints:

- C++ features
- Function overloading
- Operator overloading

4. What is function overloading? Explain briefly with program.

Hints:

- Definition for function overloading
- Syntax
- Example
- Sample program

5. Write a program to demonstrate how a static data is accessed by a static member function.

Hints:

- Describe a static variable in the class
- Use a member function as static
- Define the static member function outside the class'
- Call the static function so as to demonstrate how a static data is accessed.

6. What is friend function? What is the use of using friend functions in c++?

Explain with a program.

Hints:

- Definition of friend function
- Syntax
- Usage: non member function can access the private data of a class
- Sample program to implement friend function.

7. Discuss in detail about default arguments with an example.

Hints:

- Default argument definition
- Syntax
- The number of arguments differ in the function definition
- Sample program

UNIT II

1. Write a program to implement

a. dynamic polymorphism

Hints:

- ◆ During multiple inheritances if the appropriate member function could be selected while the program is running is known as Runtime polymorphism
- ◆ such polymorphisms are known as dynamic polymorphism
- ◆ examples
- ◆ sample programs

b. virtual function

Hints:

- Υ Different versions for the virtual function should be present in different derived classes with same name as virtual function name
- Υ Syntax
- Υ Example
- Υ Sample program

2. What is inheritance and explain briefly pointer to derived class.

Hints:

- ◆ Definition
- ◆ Types
 - Multiple inheritance
If a derived class is derived from more than one base class, then it is called multiple inheritance.

 - ◆ Hierarchical inheritance
If two or more derived classes are derived from the same base class then it is known as hierarchical inheritance

 - ◆ Multilevel inheritance
If a derived class is derived from another derived class then it is known as multilevel inheritance.

 - ◆ Sample programs

3. Explain copy constructor and destructor with suitable C++ coding.

Hints:

The **copy constructor** is a constructor which creates an object by initializing it with an object of the same class, which has been created previously.

The copy constructor is used to:

- ⇒ Initialize one object from another of the same type.
- ⇒ Copy an object to pass it as an argument to a function.

- ◆ Also write sample program with one argument

4.Explain in detail about constructor with dynamic allocation.

Hints:

- ◆ Constructor is used to initialize an object
- ◆ Types of constructor
- ◆ Dynamic constructor is used at runtime
- ◆ Syntax
- ◆ Example source codes

5.With an example explain about operator overloading through friend functions.

Hints:

In case of unary operators, overloaded operator can be invoked as

Operator op (x);

In case of binary operators, overloaded operator can be invoked as

Operator op (x , y)

- Sample program

6.Explain about Unary Operator and Binary Operator Overloading with program.

Hints:

- When unary operators are overloaded using member functions it takes no explicit arguments and return no explicit values.
- Sample program

- When binary operators are overloaded using member functions, it takes one explicit argument. Also the left hand side operand must be an object of the relevant class.
- Sample program

7.List out the rules for overloading operators with example.

Hints:

- Only the existing operators can be overloaded.
- The overloaded operator must have at least one operand that is of user defined data type
- The basic meaning of the operator should not be changed.
- Overloaded operators follow the syntax rules of the original operators. They cannot be overridden.

UNIT III

1. What is the need of Templates? Explain.

Hint:

- Templates support generic programming , which allows to develop reusable software components such as functions, classes etc., supporting different data types in a single framework
- sample program

2. What is uncaught exception function? Give an example.

Hint:

- Exception which is not executed
- Example
- Syntax with program

3. What are the use of terminate () and Unexpected functions? Explain with a program

Hint:

- Terminate is used to denote the end of exception
- Example program
- Unexpected functions are handled by exceptions as per the demonstrations

4. How to use multiple catch functions inside a program? Explain with a program.

Hints:

- Based on the definition of the multiple exception appropriate function is handled according to the choice matches
- Example explanation
- Sample program

5. Write all blocks of exception handling? Explain with a program.

Hints:

- Definition of exception handling
- Example coding
- Throw and catch
- Sample program

6. Write notes on Formatted and Unformatted Console I/O Operations.

Hints:

- Definition for formatted and unformatted operations
- Simple statements
- Statements with additional features
- Examples
- Sample programs if possible

7. Explain about File Pointers and their manipulations with example.

Hints:

- Name the file on the disk
- Open the file to get the file pointer.
- Process the file. (Read / Write)
- Check for errors while processing.
- Close the file after its complete usage.

UNIT IV

1. Explain disjoint set in detail.

Hints:

- Collection of sets where each set has a representative which is a member of the set
- Operations
- Applications
- Equivalence relations
- Find ADT

2. Explain Red Black tree operations with examples.

Hints:

- Self balancing binary search tree
- Properties
- Comparison with AVL trees
- Modifying routines
- Algorithms

3. Explain in detail about splay trees with suitable examples

Hints:

- SPLAY trees are variation of binary search trees
- Not to spend much time on balancing
- Amortized analysis
- Splaying
 - Zig zag
 - Zig zig

4. What are AVL trees? Describe the different rotations defined for the AVL tree.

Hints:

- Adelson-Velskii and Landis tree is a binary search tree except that for every node in the tree, the height of the left and right subtrees can differ by atmost 1.
- Balance factor
- Example graph
- Algorithm
- Types – single and double rotations

5. Explain in detail about the Fibonacci heaps.

Hints;

- Collection of heap ordered trees, the trees are rooted but unordered
- Pointers
- Two fields
- Examples
- Algorithm

6. Give an example for binomial heap and explain the same.

Hints:

- Definition of a binomial heap
- Properties
- Algorithm
 - Basic operations, insert,delete
 - Find minimum

UNIT V

1. Explain Graph traversals with examples.

Hints:

- Definition for traversals
- Types
- Examples
- Diagrams with algorithms

2. Describe the topological sorting method with suitable examples.

Hints:

- Definition for topological sorting
- Explanation with small array example
- C++ coding

3. What do you know about the breadth first search? Explain.

Hints:

- A type of graph traversal
- Searches on level based
- Draw a graph
- Algorithm with example

4. Write the Prim's algorithm and explain the same.

Hints:

- Definition for spanning tree
- To find the minimum spanning tree Prim's method is used
- The pair with the minimum weight is chosen
- Algorithm with small graph example

5. Explain the Kruskal's algorithm for minimum spanning tree.

Hints:

- Definition for spanning tree
- To find the minimum spanning tree Prim's method is used
- Not necessary with the minimum weight

- Circuit should not be formed
- Algorithm with small graph example

6. Discuss any two shortest path algorithms.

Hints:

- Dijkstra's Algorithm
 - Finding the distance between the start node and the neighboring nodes
 - Graph example
 - Algorithm
- Floyd's Algorithm
 - This algorithm requires a weighted graphs
 - Computes the distance matrix of a weighted graph with vertices through a series of n by n matrices.
 - Example
 - Algorithm

7. Compare depth first search and depth first search.

S.NO	DEPTH FIRST SEARCH	BREADTH FIRST SEARCH
1	Backtracking is possible from dead end	Backtracking is not possible
2	Search is done in one particular direction at the time	Vertices in the same level are maintained parallel
3	LIFO order	FIFO order

- Graph example
- Algorithm
