SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM CAMPUS, CHENNAI-89

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
18CSC202J-OBJECT ORIENTED DESIGN AND PROGRAMMING
QUESTION BANK

UNIT 4 PART A (1 Mark)

1. The STL can be used as a standard approach for----------[L1, R5-726]
a) Storing and sorting
b) **Storing and processing data**
c) data processing only
d) storing only
Ans:b

2. Name the Container which uses both stack and queue.[L1, R5-728]
a) storage
b) linked list
c) queuing
d) **Deque**
Ans:d

3. Identify the characteristics of vector container.[L2, R5-728]
a) Relocating, expandable array
b) Fixed size
c) Doubly linked list
d) link vector
Ans:a

4. Associative container uses---------to access data.[L1, R5-729]
a) queue
b) Keys
c) stack
d) string
Ans:b

5. Class templates are generally used for--------[L1, R5-690]
a) Data storage
b) debug
c) fixed data type
d) storage
Ans:a

6. In UML, Templates are also called as--------[L1, R5-702]

a) container
b) modified
c) **Parameterized**
d) generic
Ans:c

7. ___ __ specifies additional detail about UML element[**L1, R5-703]**
a) Stereotype
b) container
c) associative container
d) data processing
Ans:a

8.-------- is visible only to its containing package and to its nested package.[L1, R1-163]
a) protected
b) public
c) **Private**
d) package
Ans:c

9. Notation is used to specify the required and provided interfaces of the components. The interfaces between the components are named as------[L1, R1-172]
a) Assembly connectors
b) cooling controllers
c) Environmental controller
d) Plan analyst
Ans:a

10. List the 3 essential elements of a deployment diagram?[L1, R1-177] **a) Artifacts, nodes and connections.**
b) stack, queue, deque
c) memory, database, connections
d) package, element, deployment
Ans:a

11. Activity, use case diagram, collaboration diagram and sequence diagram are categorized as [L4, R1-147]
a) non-behavioral diagrams
b) non structural diagrams
c) structural diagrams
d) **Behavioral diagrams**
Ans:d

12. Recognize which diagram is used to distribute files, libraries and tables across topology of hardware?
[L1, R1-171]
a) Deployment diagrams

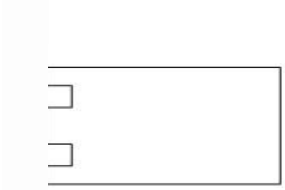
b) use case diagrams
c) sequence diagrams
d) collaboration diagrams
Ans:a

13. List the essentials in package diagram [L1, R1-165]
a) Package notation, element visibility, dependency relationship
b) package notation, sequence, dependency relationship
c) Dependency, element visibility
d) package, deployment, sequence
Ans:a

14. Good packages are---------coupled and highly cohesive among the elements in package. [L1, R1-167]
a) Tightly
b) highly
c) **loosely**
d) semi Ans:c

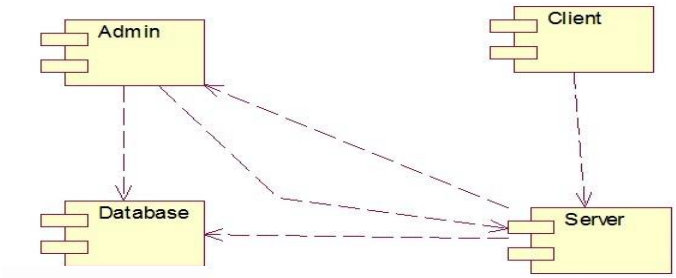
15. Identify the validity of template parameters? [L1, R5-682]
a) inside that block only
b) inside the class
c) whole program
d) inside the main class
Ans:a

16. Identify the core element of UML in the below figure?[L2, R1-163]



a)
Node
b)
Interface
c)
Class
d)
Component
Ans:d

17. Recognize the UML diagram shown below? [L2, R1-163]



a) Component
b) Deployment
c) Use Case
d) DF D
Ans:a

18. ---------- type of program can be included in try block? [L1, R5-705]
a) static memory allocation
b) Dynamic memory allocation
c) const reference
d) pointer
Ans:b

19. ---------statement is used to catch all types of exceptions. [L1, R5-705]
a)
catch()
b) catch(Test)
c)
catch(…)
d) none of the mentioned
Ans:c

20. The ---------- class name must be included in the class in which it is located. [L1, R5-709]
a) try
b) Exception
c) catch
d) template
Ans:b

21. From where does the template class derived? [L2, R5-702]

1. Regular non-templated C++ class
2. Templated class
3. A or B
4. None of the above

a) Only 1st is correct
b) Only 2nd is correct
c) Both 1st and 2nd are correct
d) **Maybe 1st or 2nd is correct.**
Ans:d

22. Explore the correct statement about string template?[L3, R5-683]
a) It is used to replace a string.
b) **It is used to replace a string with another string at runtime.**
c) It is used to delete a string.
d) None of the above
Ans:b

23. Identify which among the following is not correct. [L2, R5-683]
a) **template <class T> func(T x)**
{}
template <class T> func<T*>(T* x) {}
b) template <class T
class myObject {};
c) template <class T>
class myObj { template <class R> memFunc() {} };
d) All of the above are correct.
Ans:a

24. Examine whether templates are conceptually related to polymorphism? [L4, R5-682]
a) Not Related
b) Only when the template types are objects
c) **Yes, but compile-time polymorphism**
d) Yes, but run-time polymorphism
Ans:c

25. Identify an invalid template declaration.[L2, R5-683]
a) template <int x> int func() {return x;}
b) **template <double x> double func() {return x;}**
c) template <typename x> void func(x t) {}
Ans:b
d) It is not possible in CPP to restrict a function

26. Relate an option to restrict a function to throw certain exceptions? [L3, R5-705]

a) Defining multiple try and catch block inside a function

b) Defining generic function within try block

c) Defining function with throw clause

Ans:c

27. Select the ways to represent nodes in a deployment diagram? [L6, R1-171]

a) Nodes instances are underlined identifiers of the form name:type

b) The name may be left off, indicating an unnamed instance of the type

c) The type may be left off, indicating a named instance with an unspecified type

d) **All of the mentioned**

Ans:d

28. Examine: In component diagrams, building block which is represented with two rectangles laid on left side is classified as [L4, R1-163]

a) type of components

b) interfaces

c) **dependency relationships**

d) State dependency

Ans:c

29. Choose the ways to represent nodes in a deployment diagram? [L6, R1-171]

a) Nodes instances are underlined identifiers of the form name:type

b) The name may be left off,indicating an unnamed instance of the type

c) The type may be left off, indicating a named instance with an unspecified type

d) **All of the mentioned**

Ans:d

30. Identify the ways to represent nodes in a deployment diagram? [L2, R1-171]

a. Nodes instances are underlined identifiers of the form name:type

b. The name may be left off, indicating an unnamed instance of the type

c. The type may be left off, indicating a named instance with an unspecified type

d. **All of the mentioned**

Ans:d

PART- B (4 Marks)

1. What do you meant by Generic Programming? What are its advantages and state few applications.
2. Define a Class Template. Write a suitable example program.
3. What is a Function Template? Illustrate with a suitable example program.

4. Give the differences between Class template and Function template.
5. What do you meant by Overloaded Function Template? What are the rules to be followed to select a suitable template.
6. Distinguish between overloaded functions and function templates
7. What is the need for template function in C++? What are the advantages?
8. What is an exception? How it is handled in C++?
9. Write a program to demonstrate the concept of rethrowing an exception.
10. What are the two kinds of exception?
11. Illustrate multiple catch statements with a suitable example.
12. When should a function throw an exception? Give an example to illustrate it.
13. What is uncaught_exception() function? why do we need it?
14. When do we need multiple catch Handlers? Give an example.
15. What are standard exceptions? List the types of exception and specify the position when it was generated?
16. What is a component diagram and state its artifacts.
17. Define Deployment diagram. State the artifacts to be identified before drawing a Deployment diagram.
18. What are the uses of the Component diagram and Deployment diagram?
19. Give the notations of Component diagram and Deployment Diagram
20. Define a Package Diagram. Give the advantages of using a Package diagram.

PART C (12 Marks)

Answer the Questions Briefly:-

1. Explain the two models for template compilation? Compare.
2. Discuss in detail on Class Template with a suitable example.
3. Illustrate with an example how a template class can be used as a base class.
4. What is a Function Template? Discuss in detail with a suitable program.
5. Write a program which generate a template class by which one can perform integer type data addition and float type data addition.
6. Give the syntax for function template. Write template function for computing the bubble sort. Write a test program to illustrate its use.
7. Discuss in detail on Overlading Function template. Illustrate it with a suitable program.
8. Explain how the Class Template can be used with Operator overloading with a program.
9. Write a C++ program using function template to find the area of a square, rectangle and traingle.
10. Write a C++ program to develop a Simple Calculator to perform arithmetic operations using Class Template.
11. What is an exception? How it is handled in C++ programs? Explain how the control is transferred when exceptions occur during programs execution. Write a program to illustrate exception handling.
12. Write a program to show how to restrict the types of exceptions that can be thrown by a function.
13. Write a program to show how to rethrow an exception.

14. Using time class, throw an exception when invalid time is input, write set_terminate() to provide your own terminate function which takes care of this problem.
15. Write a C++ program to demonstrate the use of try, catch , throw and nested try.
16. What is a user defined exception. Write down the scenario where we require user defined exceptions.
17. When do we need multiple catch blocks for a single try block? Write a program to illustrate it.
18. Explain in detail on the use of Multiple catch statements in a program with a suitable example. Discuss the importance of “catch all exception” with a program.
19. Discuss in detail on Component Diagram with a suitable example.
20. Explain Deployment Diagram with a suitable example.
21. What is a Package Diagram. Illustrate it with a suitable example.
22. Draw the Component, Deployment and package diagram for the ATM Banking system. Explain the system with the notations used in each diagram.
23. Design the Component, Deployment and package diagram for the Airline Reservation system. Explain the system with the notations used in each diagram.
24. Explain the Component, Deployment and package diagram for the Course Registration system with a neat diagram Illustrate the system with the notations used in each diagram.