SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai - 600089

**FACULTY OF ENGINEERING AND TECHNOLOGY**

# DEPARTMENT OF INFORMATION TECHNOLOGY

****

**QUESTION BANK**

**DEGREE / BRANCH: B.Tech-IT**

**III SEMESTER**

**SUB CODE – SUBJECT NAME**

**18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING**

**Regulation 2018**

**Academic Year 2022-23 (ODD SEMESTER)**

# SRMINSTITUTE OF SCIENCE AND TECHNOLOGY

**Ramapuram Campus, Bharathi Salai, Ramapuram, Chennai-600089**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**QUESTION BANK**

**SUBJECT : 18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING**

**SEM/YEAR: III/II**

**Course Outcomes**

*CO-*1: Identify the class and build domain model

CO-2 : Construct programs using method overloading and operator overloading

CO-3 : Create programs using inline, friend and virtual functions, construct programs using standard templates

CO-4 : Construct programs using exceptional handling and collections

CO-5 : Create UML component diagram and deployment diagram

CO-6 : Create programs using object oriented approach and design methodologies

| **UNIT I** | | | | |
| --- | --- | --- | --- | --- |
| Comparison of Procedural and Object Oriented Programming- OOPS and its features - I/O Operations, Data Types, Variables, static - Constants, Pointers, Type Conversions - Features: Class and Objects - UML Diagrams Introduction- Feature :Class and Objects - Examples of Class and Objects - UML Class Diagram and its components - Class Diagram relations and Multiplicity - Feature Abstraction and Encapsulation - Application of Abstraction and  Encapsulation - Access specifiers – public, private - Access specifiers - protected, friend, inline - UML use case Diagram, use case, Scenario - Use case Diagram objects and relations - Method, Constructor and Destructor - Method, Constructor and Destructor | | | | |
| **PART-A (Multiple Choice Questions)** | | | | |
| **Q.**  **No** | **Questions** | **Course Outcome** | **Competence**  **BT Level** | |
| **1** | **Which of the following explains Polymorphism?**  A) intfunc(int, int);  Float func1(float,float);  B) intfunc(int);  Intfunc(int);  **C) intfunc(float); Intnew\_func();**  D) intfunc(); Intnew\_func(); | CO1 | 2 | |
| 2 | Find how many bytes are occupied by the following data types in a 32-bit system.  A) Type int  B) Type long double  **C) Type float**  D)Type long | CO1 | 4 | |
| **3** | Which of the following is a important role of a function?  A. give a name to a block of code  B, reduce program size  C. accept arguments and provide a return value  **D. help organize a program into conceptual units** | CO1 | 4 | |
| **4** | The Unified Modeling Language is ………………………   1. A. a program that builds physical models. 2. B. a way to look at the organization of a program 3. C. the combination of C++ and FORTRAN 4. **D. helpful in developing software systems.** | CO1 | 1 | |
| **5** | In C++, a function contained within a class is called ……………….   1. **A. a member function** 2. B. an operator 3. C. a class function 4. D. a method | CO1 | 1 | |
| **6** | What happens if the base and derived class contains definition of a function with same prototype?   1. A. Compiler reports an error on compilation 2. B. Only base class function will get called irrespective of object 3. C. Only derived class function will get called irrespective of object 4. **D. Base class object will call base class function and derived class object will call derived class object will call derived class function** | CO1 | 4 | |
| **7** | Which one of the following option is correct about the statement given below? The compiler checks the type of reference in the object and not the type of object   1. A. inheritance 2. **B. Polymorphism** 3. C. Abstraction 4. D. Encapsulation | CO1 | 4 | |
| **8** | Which of the following functions are performed by a constructor?  A) Construct a new class  B) Construct a new object  C) Construct a new function  **D) Initialize objects** | CO1 | 4 | |
| **9** | Which of the following is the correct class of the object cout  A) iostream  B) istream  **C) ostream**  D) ifstream | CO1 | 2 | |
| **10** | In UML, diagrams which captures system static structure and provide foundation for other models is called ...............  A) Deployment diagram  **B) Class diagram**  C) Component diagram  D) Object diagram | CO1 | 2 | |
| **11** | Find the error produced by compiler when private members are accessed?  **A) Can’t access private message**  B) Code unreachable  C) Core dumped  D) Bad code | CO1 | 4 | |
| **12** | Choose the default access specifier for the class member  A) public   1. **B) private** 2. C) protected 3. D) None of the above | CO1 | 1 | |
| **13** | Which of the following is CPP style type-casting?  A) per=total/ (float)m  **B) per=total/float(m)**  C) per = (float)total/m  D) None of these | CO1 | 1 | |
| **14** | What is the output of the following program?  #include<iostream>  using namespace std;  void main()  {  char s() = “SRM”;  \*s = ‘R’;  cout<<s<<endl;  }  **A) RRM**  B) SRM  C) SRR  D) None of these | CO1 | 4 | |
| **15** | What does the following statement mean?  int (\*fp)(char\*)   1. A) pointer to a pointer 2. B) pointer to an array of chars 3. **C) pointer to function taking a char\* argument and returns an int** 4. D) function taking a char\* argument and returning a pointer to int | CO1 | 4 | |
| **16** | Which of the following concepts of OOPS means exposing only necessary information to client?   1. A) Encapsulation 2. **B) Abstraction** 3. C) Polymorphism 4. D) Data binding | CO1 | 2 | |
| **17** | Which of the following is illegal?   1. A) int \*ip; 2. B) string s, \*sp = 0;   **C) int i; double\* dp = &i;**   1. D) int \*pi = 0; | CO1 | 4 | |
| **18** | Which member can never be accessed by inherited classes?  **A) Private member function**   1. B) Public member function 2. C) Protected member function 3. D) All can be accessed | CO1 | 1 | |
| **19** | Analyze the code and choose the correct  int a=100, b=200;  int \*p=&a, \*q=&b;  p=q;   1. A) b is assigned to a 2. **B) p now points to b** 3. C) a is assigned to b 4. D) q now points to a | CO1 | 4 | |
| **20** | Mention the size\_t integer type in C++ is?   1. A) Unsigned integer of at least 64 bits 2. B) Signed integer of at least 16 bits 3. **C) Unsigned integer of at least 16 bits** 4. D) Signed integer of at least 64 bits | CO1 | 2 | |
| **21** | Which among the following is not a property of an object?  a) Identity b) Properties c) Attributes **d) Names** | CO1 | 1 | |
| **22** | Which is most appropriate comment on following class definition?  class Student  {  int a;  public : float a;  };   1. **Error : same variable name can’t be used twice** 2. Error : Public must come first 3. Error : data types are different for same variable 4. It is correct | CO1 | 5 | |
| **23** | Instance of which type of class can’t be created?  a) Anonymous class b) Nested class c) Parent class **d) Abstract class** | CO1 | 2 | |
| **24** | What is the output of following code?  int n=10; // global  class A()  {  private : int n;  public : int m;  A()  {  n=100; m=50;  }  void disp()  { cout<<”n”<<m<<n;  };  a) 1050100 b) 1005010 c) n5010 **d) n50100** | CO1 | 5 | |
| **25** | Find the output of the following program.  class education  {  char name[10];  public : disp()  {  cout<<”Its education system”;  }  class school:public education  {  public: void dsip()  {  cout<<”Its school education system”;  }  };  void main()  {  school s;  s.disp();  }  }  **a) Its school education system** b) Its education system c) Its school education systemIts education system d) Its education systemIts school education system | CO1 | 5 | |
| **PART B (4 Marks)** | | | | |
| **1** | How can we prevent a class from instantiation? | CO1 | | 2 |
| **2** | Construct Use-case diagram for an Online Shopping Application | CO1 | | 3 |
| **3** | List out the difference between procedure oriented programming & Object oriented programming | CO1 | | 1 |
| **4** | Write syntax of class, objects and methods and explain with example. | CO1 | | 2 |
| **5** | Write a C++ program to generate factorial of a number using class | CO1 | | 3 |
| **6** | Write an example program to demonstrate type conversions and explain | CO1 | | 2 |
| **7** | Consider a Banking System. Identify three entities in the system which can be represented using classes and show the relationship between them using UML class diagrams | CO1 | | 6 |
| **8** | What is a constructor? What are its uses? | CO1 | | 2 |
| **PART C (12 Marks)** | | | | |
| **1** | Write the problem statement for Library Management system. Design UML Class diagram and explain its components | CO1 | | 6 |
| **2** | There are 50 computers available in computer programming lab where each computers are used six hours per day. Write a C++ program using classes and objects that contain getDetail() for getting input from user, calculate second per Day() for calculating the usage of each computer in seconds per day ,calculateminutesperWeek() for calculating the usage of each computer in minutes per week ,calculatehourperMonth() for calculating usage of each computer in hour per month and calculatedayperYear() for calculating usage of each computer in day per year | CO1 | | 6 |
| **3** | Give example for cast? Explain OOPS features with suitable example. | CO1 | | 2 |
| **4** | A University conducts examinations and the results are announced. Prepare a report for the following:  • Print the marks in the register number order semester wise for each department  • Print the Arrear list semester wise.  • Prepare a Rank list for each department. | CO1 | | 3 |
| **5** | Create three classes with names Shape, Rectangle and Circle and make use of the functions getdata(), printdata(), and area(). To find the area of circle and rectangle, which type of inheritance is suitable? Why? Explain? | CO1 | | 6 |

| **UNIT II** | | | |
| --- | --- | --- | --- |
| Types of constructor (Default, Parameter), Static and copy constructor - Feature Polymorphism: Constructor overloading, Method Overloading - Example for method overloading, Method Overloading: Different parameter with different return values - Operator Overloading and types, Overloading Assignment Operator - Overloading Unary Operators, Example for Unary Operator overloading - Overloading Binary Operators, Example for Binary Operator overloading - UML Interaction Diagrams, Sequence Diagram - Collaboration Diagram, Example Diagram - Feature: Inheritance, Inheritance and its types | | | |
| **PART-A (Multiple Choice Questions)** | | | |
| **Q.**  **No** | **Questions** | **Course Outcome** | **Competence**  **BT Level** |
| **1** | While overloading binary operators using member function, it requires \_\_\_ arguments.  a. Zero  b. **One**  c. Two  d. Three | CO2 | BT1 |
| **2** | Which of the followings are true about constructors?  A. A class can have more than one constructor.  B. They can be inherited.  C. Their address can be referred.  D. Constructors cannot be declared in protected section of the class.  E. Constructors cannot return values.  a. Only A, B, D  b. A,B,D,E  c. A,C,E  d. **A,D,E** | CO2 | BT1 |
| **3** | Which of the following keyword is used to overload an operator?  a. overload  b. **operator**  c.friend  d.overrider | CO2 | BT1 |
| **4** | What will happen if a class is not having any name?  a. it cannot have a destructor  b. It cannot have a constructor.  c. It is not allowed.  d. **Both A and B** | CO2 | BT1 |
| **5** | Which inheritance type is used in the class given below?  class A : public X, public Y  a. Multilevel inheritance  b. **Multiple inheritance**  c. Hybrid inheritance  d .Hierarchical Inheritance | CO2 | BT1 |
| **6** | Which of the following operators cannot be overloaded?  a. []  b. ->  c. **?:**  d. \* | CO2 | BT1 |
| **7** | In which of the following a virtual call is resolved at the time of compilation?  a. From inside the destructor.  b. From inside the constructor.  c. From inside the main ().  d. **Both A and B** | CO2 | BT2 |
| **8** | Which of the following operator is overloaded for object cout?  a. >>  **b. <<**  c. +  d. = | CO2 | BT1 |
| **9** | Assume class TEST. Which of the following statements is/are responsible to invoke copy constructor?  a. TEST T2 (T1)  b. TEST T4 = T1  c. T2=T1  d. **Both A and B** | CO2 | BT2 |
| **10** | Which of the following is the perfect set of operators that can’t be overloaded in CPP?  a. +=, ?, :: , >>  b. >>, <<, ?,\*,sizeof()  **c. ::, ., .\*, ?:**  d. ::, ->, \*, new delete | CO2 | BT2 |
| **11** | How many operators are supported by C++?  a. 30 operators  b. 40 operators  **c. 45 operators**  d. 65 operator | CO2 | BT1 |
| **12** | A non-member function that is given access to all members of a class within it is declared, is called  a. Access function  **b. Friend function**  c. Operator functions  d. None of them | CO2 | BT1 |
| **13** | Which of the following operators should be preferred to overload as a global function rather than a member method?  a. Postfix ++  b. Comparison Operator  c. **Insertion Operator <<**  d. Prefix++ | CO2 | BT1 |
| **14** | We can overload which of the following C++ operators  a. **Arithmetic operator (+, -, \*, /)**  b. Class Member Access Operators (., .\*)  c. Size operator (sizeof)  d. Conditional operator (?:) | CO2 | BT2 |
| **15** | Operator overloading is also called …………….. polymorphism  a. run time  b. initial time  c. **Compile time**  d. Completion time | CO2 | BT1 |
| **16** | Operator overloading is done with the help of a special function called ……………, which describes the special task of an operator.  a. overloading function  b. special task function  c. detail function  d. **operator function** | CO2 | BT1 |
| **17** | Overload an operator by naming it a  a. variable  b. built-in type  c. **function**  d. class | CO2 | BT1 |
| **18** | Which of the function operator cannot be over loaded  a. <=  b. **?:**  c. ==  d. \* | CO2 | BT2 |
| **19** | Kind of diagrams which are used to show interactions between series of messages are classified as  a. activity diagrams  b. state chart diagrams  **c. collaboration diagrams**  d. object lifeline diagrams | CO2 | BT1 |
| **20** | Dynamic aspects related to a system are shown with help of  a. sequence diagrams  b. **interaction diagrams**  c. deployment diagrams  d. use case diagrams | CO2 | BT1 |
| **21** | Determine which diagrams are used to show interactions between series of messages  a.Activity diagrams  b. State Chart diagrams  c. **Collaboration diagrams**  d. Object Lifeline diagrams | CO2 | BT1 |
| **22** | Identify the syntax of overloading operator + for class A?  **a.A operator + (arg\_list){}**  b. A operator [+] (arg\_list){}  c. int +(arg\_list){}  d. int [+](arg\_list){} | CO2 | BT1 |
| **23** | Classify three different types of message arrows   1. Synchronous, asynchronous, asynchronous with instance creation 2. Self, multiplied, instance generator 3. **Synchronous, asynchronous, synchronous with instance creation** 4. None of the above | CO2 | BT1 |
| **24** | **Which feature of OOP indicates code reusability?** a. Abstraction b. Polymorphism c. Encapsulation d. **Inheritance** | CO2 | BT1 |
| **25** | **For constructor overloading, each constructor must differ in \_\_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_** a. **Number of arguments and type of arguments** b. Number of arguments and return type c. Return type and type of arguments d. Return type and definition | CO2 | BT1 |
| **PART B (4 Marks)** | | | |
| **1** | What is the necessity of constructor overloading? | CO2 | BT2 |
| **2** | Categorize the types of Constructors | CO2 | BT1 |
| **3** | Define method overloading. Write a program to implement method overloading with different number of arguments and same return types | CO2 | BT3 |
| **4** | Write down the restrictions on Operator overloading | CO2 | BT2 |
| **5** | Can we have virtual destructors? If so what is the use of virtual destructors. | CO2 | BT3 |
| **6** | Define collaboration diagram with its notation | CO2 | BT1 |
| **7** | Explain the modes of inheritance with an example | CO2 | BT2 |
| **8** | Judge the output of the following C++ code?  #include <iostream>  using namespace std;  class Integer  { int i;  public: Integer(int ii) : i(ii) {}  const Integer  operator+(const Integer&rv) const  {  cout<< "operator+" <<endl;  return Integer(i + rv.i);  }  Integer&  operator+=(const Integer&rv)  {  cout<< "operator+=" <<endl;  i += rv.i; return \*this;  } };  int main()  {  int i = 1, j = 2, k = 3;  k += i + j;  Integer ii(1), jj(2), kk(3); kk += ii + jj;  } | CO2 | BT4 |
| **9** | Relate the differences between constructors and destructors | CO2 | BT2 |
|  | | | |
| **10** | Define overloading unary and binary operators with an example | CO2 | BT2 |
| **PART C (12 Marks)** | | | |
| **1** | Define UML Sequence diagram with its notations. Draw the sequence diagram for Online hotel management | CO2 | BT4 |
| **2** | Write a C++ program for Constructor overloading (parameterized, default and copy) | CO2 | BT3 |
| **3** | Define Sequence and Collaboration diagram. Draw the sequence and collaboration diagram for ATM amount withdrawal | CO2 | BT4 |
| **4** | Define interaction diagram. Draw an Sequence diagram for online hospital management | CO2 | BT4 |
| **5** | Explain in detail about operator overloading and its types with example | CO2 | BT3 |

| **UNIT III** | | | |
| --- | --- | --- | --- |
| Feature Inheritance: Single and Multiple, Inheritance: Multilevel, Hybrid, Hierarchial-Advanced Functions: Inline, Friend- Advanced Functions: Virtual, Overriding- Advanced Function: Pure Virtual function- Example for Virtual and pure virtual function- Abstract class and Interface- UML State Chart Diagram- UML Activity Diagram | | | |
| **PART-A (Multiple Choice Questions)** | | | |
| **Q.**  **No** | **Questions** | **Course Outcome** | **Competence**  **BT Level** |
| **1** | **Diagrams which are used to distribute files, libraries and tables across topology of hardware are called?**  a) deployment diagrams  b) use case diagrams  c) sequence diagrams  d) collaboration diagrams  Ans: d | C04 | BT 1 |
| **2** | **Name the function whose definition can be substituted at a place where its function call is made: ?**  a) friends function  b) inline function  c) volatile function  d) external function  Ans: b | C04 | BT 2 |
| **3** | **Activity diagram, use case diagram, collaboration diagram and sequence diagram are considered**  **as types of ?**  a) non-behavioral diagrams  b)nonstructural diagrams  c) structural diagrams  d) behavioral diagrams  Ans: d | C04 | BT 2 |
| **4** | **Use of pointers or reference to an abstract class gives rise to which among the following feature?**  a) Static Polymorphism  b) Runtime polymorphism  c) Compile time Polymorphism  d) Polymorphism within methods  Ans: b | C04 | BT 3 |
| **5** | **Which diagram in UML shows a complete or partial view of the structure of a modeled system at a specific time?**  a) Sequence Diagram  b) Collaboration Diagram  c) Class Diagram  d) Object Diagram  Ans: d | C04 | BT 2 |
| **6** | **Can abstract class have main () function defined inside it?**  a) Yes, depending on return type of main()  b) Yes, always  c) No, main must not be defined inside abstract class  d) No, because main() is not abstract function  Ans: b | C04 | BT 1 |
| **7** | **If there is an abstract method in a class then, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  a) Class must be abstract class  b) Class may or may not be abstract class  c) Class is generic  d) Class must be public  Ans: a | C04 | BT 1 |
| **8** | **Which of the following UML diagrams has a static view?**  a) Collaboration  b) Use case  c) State chart  d) Activity  Ans: b | C04 | BT 1 |
| **9** | **11. Which keyword is used to declare the friend function?**  a) friend  b) friend  c) classfriend  d) myfriend  Ans: b | C04 | BT 2 |
| **10** | **Which of the following cannot be used to declare a class as a virtual?**  a) Methods  b) Properties  c) Events  d) Fields  Ans: d | C04 | BT 2 |
| **11** | **Which of the given modifiers can be used to prevent Method overriding?**  a) Static  b) Constant  c) Sealed  d) final  Ans: c | C04 | BT 1 |
| **12** | **Which problem arises due to multiple inheritances, if hierarchical inheritance is used previously for**  **its base classes?**  a) Diamond  b) Circle  c) Triangle  d) Loop  Ans: a | C04 | BT 2 |
| **13** | **How many classes should a program contain to implement the multiple inheritance?**  a) Only 1  b) At least 1  c) At least 3  d) Exactly 3  Ans: c | C04 | BT 1 |
| **14** | **How many basic types of inheritance are provided as OOP feature?**  a) 4  b) 3  c) 2  d) 1  Ans: a | C04 | BT 1 |
| **15** | **How can you make the private members inheritable?**  a) By making their visibility mode as public only  b) By making their visibility mode as protected only  c) By making their visibility mode as private in derived class  d) It can be done both by making the visibility mode public or protected  Ans: d | C04 | BT 1 |
| **16** | **Which access type data gets derived as private member in derived class?**  a) Private  b) Public  c) Protected  d) Protected and Private  Ans: a | C04 | BT 1 |
| **17** | **Which programming language doesn’t support multiple inheritances?**  a) C++ and Java  b) C and C++  c) Java and Small Talk  d) Java  Ans: d | C04 | BT 2 |
| **18** | **Which among the following best defines single level inheritance?**  a) A class inheriting a derived class  b) A class inheriting a base class  c) A class inheriting a nested class  d) A class which gets inherited by 2 classes  Ans: b | C04 | BT 2 |
| **19** | **Is it compulsory to have constructor for all the classes involved in multiple inheritance?**  a) Yes, always  b) Yes, only if no abstract class is involved  c) No, only classes being used should have a constructor  d) No, they must not contain constructors  Ans: b | C04 | BT 2 |
| **20** | **Can the derived class be made abstract if multiple inheritances is used?**  a) No, because other classes must be abstract too  b) Yes, if all the functions are implemented  c) Yes, if all the methods are predefined  d) No, since constructors won’t be there  Ans: d | C04 | BT 2 |
| **21** | **Which keyword is used to declare the friend function?** a) firend b) friend c) classfriend d) myfriend  Ans:d | C04 | BT 1 |
| **22** | **Pick out the correct option.** a) We cannot make an instance of an abstract base class b) We can make an instance of an abstract base class c) We can make an instance of an abstract super class d) We can make an instance of an abstract derived class  Ans:a | C04 | BT 1 |
| **23** | **Which is used to create a pure virtual function?** a) $ b) =0 c) & d) !  Ans:b | C04 | BT 3 |
| **24** | **Which is also called as abstract class?** a) virtual function b) pure virtual function c) derived class d) base class  Ans:b | C04 | BT 3 |
| **25** | **What is the syntax of friend function?** a) friend class1 Class2; b) friend class; c) friend class d) friend class()  Ans:a | C04 | BT3 |
| **PART B (4 Marks)** | | | |
| **1** | Explain Virtual Function | CO2 | BT2 |
| **2** | What is pure virtual function? | CO2 | BT2 |
| **3** | Explain the notations of Activity Diagram and Statechart Diagram | CO4 | BT3 |
| **4** | What is Inheritance?Benefits of Inheritance? | CO2 | BT2 |
| **5** | Describe the types of inheritance? | CO2 | BT3 |
| **6** | Difference between Virtual Function and Pure Virtual Function. | CO2 | BT2 |
| **7** | Explain about Abstract Class and Interface. | CO3 | BT2 |
| **8** | What is friend function and friend class? | CO3 | BT2 |
| **PART C (12 Marks)** | | | |
| **1** | Draw UML state chart and Activity Diagram for ATM Machine | CO4 | BT4 |
| **2** | Describe in detail about advanced friend function and friend class with example? | CO2 | BT2 |
| **3** | Describe Multilevel Inheritance with example program? | CO2 | BT3 |
| **4** | Explain Pure Virtual Function with example program? | CO3 | BT2 |
| **5** | Draw UML state chart and Activity Diagram for Library Management System | CO4 | BT4 |

| **UNIT IV** | | | |
| --- | --- | --- | --- |
| Generic – Templates : Introduction - Function Template – Example programs : Function Templates - Class Template – Example programs : Class Templates – Exceptional Handling : try and catch – Multilevel exceptional – throe and throws – finally – User defined exception – Example programs using C++ - Dynamic modelling : Package Diagram – UML Component diagram – Deployment Diagram – Example : Package, Deployment, Component diagram. | | | |
| **PART-A (Multiple Choice Questions)** | | | |
| **Q.**  **No** | **Questions** | **Course Outcome** | **Competence**  **BT Level** |
| **1** | **The STL can be used as a standard approach for-----------**  a) Storing and sorting  **b) Storing and processing data**  c) data processing only  d) storing only | CO3 | 1 |
| **2** | **Name the Container which uses both stack and queue.**  a) storage  b) linked list  c) queuing  d) **Deque** | CO3 | 1 |
| **3** | **Identify the characteristics of vector container.**  **a) Relocating, expandable array**  b) Fixed size  c) Doubly linked list  d) link vector | CO3 | 2 |
| **4** | **Associative container uses---------to access data.**  a) queue  **b) Keys**  c) stack  d) string | CO3 | 1 |
| **5** | **Class templates are generally used for--------**  **a) Data storage**  b) debug  c) fixed data type  d) storage | CO3 | 1 |
| **6** | **In UML, Templates are also called as**--------  a) container  b) Modified  **c) Parameterized**  d) generic | CO3 | 1 |
| **7** | **Identify the validity of template parameters?**  **a) inside that block only**  b) inside the class  c) whole program  d) inside the main class | CO3 | 2 |
| **8** | **Identify which among the following is not correct.**  a) **template <class T> func(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 | CO3 | 2 |
| **9** | **Examine whether templates are conceptually related to polymorphism?**  a) Not Related  b) Only when the template types are objects  **c) Yes, but compile-time polymorphism**  d) Yes, but run-time polymorphism | CO3 | 2 |
| **10** | **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) {}  d) It is not possible in CPP to restrict a function | CO3 | 2 |
| **11** | **Explore the correct statement about string template?**  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 | CO3 | 2 |
| **12** | \_\_\_\_\_\_\_\_\_\_\_t**ype of program can be included in try block?**  a) static memory allocation  b) **dynamic memory allocation**  c) const reference  d) pointer | CO4 | 1 |
| **13** | **---------statement is used to catch all types of exceptions.**  a) catch()  b) catch(Test t)  c) **catch(…)**  d) no one of the mentioned | CO4 | 1 |
| **14** | **The ------------ class name must be included in the class in which it is**  **located.**  a) try  b) **Exception**  c) catch  d) template | CO4 | 1 |
| **15** | **Select the ways to represent nodes in a deployment diagram?**  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** | CO5 | 1 |
| **16** | **\_\_\_ \_\_ specifies additional detail about UML element**.  **a) Stereotype**  b) container  c) associative container  d) data processing | CO5 | 1 |
| **17** | **-------- is visible only to its containing package and to its nested**  **a) package.**  b) protected  c) public  **d) Private** package | CO5 | 1 |
| **18** | **Notation is used to specify the required and provided interfaces of**  **the components. The interfaces between the components are named**  **as----**  **a) Assembly connectors**  b) cooling controllers  c) Environmental controller  d) Plan analyst | CO5 | 1 |
| **19** | **List the 3 essential elements of a deployment diagram.**  **a) Artifacts, nodes and connections.**  b) stack, queue, deque  c) memory, database, connections  d) package, element, deployment | CO5 | 1 |
| **20** | **Activity, use case diagram, collaboration diagram and sequence**  **diagram are categorized as** \_\_\_\_\_\_\_\_\_\_\_\_\_\_.  a) non-behavioral diagrams  b) non structural diagrams  c) structural diagrams  **d) Behavioral diagrams** | CO5 | 1 |
| **21** | **Recognize which diagram is used to distribute files, libraries and**  **tables across topology of hardware?**  **a) Deployment diagrams**  b) use case diagrams  c) sequence diagrams  d) collaboration diagrams | CO5 | 2 |
| **22** | **List the essentials in package diagram**  **a) Package notation, element visibility, dependency relationship**  b) package notation, sequence, dependency relationship  c) Dependency, element visibility  d) package, deployment, sequence | CO5 | 1 |
| **23** | **Good packages are---------coupled and highly cohesive among the elements in package.**  a) Tightly  b) highly  **c) loosely**  d) semi | CO5 | 1 |
| **24** | **Identify the core element of UML in the below figure?**    a)Node  b) Interface  c) Class  **d) Component** | CO5 | 2 |
| **25** | **Recognize the UML diagram shown below?**    **a) Component**  b) Deployment  c) Use case  d) DFD | CO5 | 2 |
| **PART B (4 Marks)** | | | |
| **1** | What do you mean by Generic Programming? What are its advantagesand state few applications? | CO3 | 1 |
| **2** | Define a Class Template. Write a suitable example program. | CO3 | 2 |
| **3** | What is a Function Template? Illustrate with a suitable example program. | CO3 | 2 |
| **4** | What do you mean by Overloaded Function Template? What are the rules to be followed to select a suitable template? | CO3 | 1 |
| **5** | Distinguish between overloaded functions and function templates | CO3 | 2 |
| **6** | What is the need for template function in C++? What are the advantages? | CO3 | 1 |
| **7** | Give the differences between Class template and Function template. | CO3 | 2 |
| **8** | What is an exception? How it is handled in C++? | CO4 | 1 |
| **9** | Write a program to demonstrate the concept of rethrowing an exception. | CO4 | 3 |
| **10** | What are the two kinds of exception? | CO4 | 1 |
| **11** | Illustrate multiple catch statements with a suitable example. | CO4 | 2 |
| **12** | When should a function throw an exception? Give an example to illustrate  it. | CO4 | 2 |
| **13** | What is uncaught\_exception() function? why do we need it? | CO4 | 2 |
| **14** | When do we need multiple catch Handlers? Give an example. | CO4 | 2 |
| **15** | What are standard exceptions? List the types of exception and specify the  position when it was generated? | CO4 | 1 |
| **16** | What is a component diagram and state its artifacts. | CO5 | 1 |
| **17** | Define Deployment diagram. State the artifacts to be identified before  drawing a Deployment diagram. | CO5 | 1 |
| **18** | What are the uses of the Component diagram and Deployment diagram? | CO5 | 2 |
| **19** | Give the notations of Component diagram and Deployment Diagram | CO5 | 1 |
| **20** | Define a Package Diagram. Give the advantages of using a Package  diagram. | CO5 | 1 |
| **PART C (12 Marks)** | | | |
| **1** | Discuss in detail on Class Template with a suitable example. | CO3 | 2 |
| **2** | What is a Function Template? Discuss in detail with a suitable program. | CO3 | 2 |
| **3** | Write a program which generate a template class by which one can perform  integer type data addition and float type data addition. | CO3 | 3 |
| **4** | Discuss in detail on Overlading Function template. Illustrate it with a  suitable program. | CO3 | 2 |
| **5** | Explain how the Class Template can be used with Operator overloading  with a program. | CO3 | 2 |
| **6** | Write a C++ program to develop a Simple Calculator to perform arithmetic  operations using Class Template. | CO3 | 3 |
| **7** | 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. | CO4 | 2 |
| **8** | Write a program to show how to restrict the types of exceptions that can be  thrown by a function. | CO4 | 3 |
| **9** | Write a program to show how to rethrow an exception. | CO4 | 3 |
| **10** | Write a C++ program to demonstrate the use of try, catch , throw and nested  try. | CO4 | 3 |
| **11** | What is a user defined exception. Write down the scenario where we  require user defined exceptions. | CO4 | 2 |
| **12** | When do we need multiple catch blocks for a single try block? Write a  program to illustrate it. | CO4 | 2 |
| **13** | 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. | CO4 | 2 |
| **14** | Explain Deployment Diagram with a suitable example. | CO5 | 2 |
| **15** | What is a Package Diagram. Illustrate it with a suitable example. | CO5 | 2 |
| **16** | Draw the Component, Deployment and package diagram for the ATM  Banking system. Explain the system with the notations used in each  diagram. | CO5 | 3 |
| **17** | Design the Component, Deployment and package diagram for the Airline  Reservation system. Explain the system with the notations used in each  diagram. | CO5 | 3 |
| **18** | 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. | CO5 | 3 |
| **19** | Discuss in detail on Component Diagram with a suitable example. | CO5 | 2 |
| **20** | Draw the Component, Deployment and package diagram for the Online  Shopping system. Explain the system with the notations used in each  diagram. | CO5 | 3 |

| **UNIT V** | | | | | |
| --- | --- | --- | --- | --- | --- |
| STL: Containers, sequence and Associative containers, Sequence Containers : Vector, List, Deque , Array, STL - stack -Associative Containers - Map - MultiMap - Iterator and Specialized Iterator - Functions of Iterator - Algorithms -find() -count() - sort() - search() - merge() -function object: for\_each, transform - streams and Files :introduction - classes and errors- disk file handling -reading and writing | | | | | |
| **PART-A (Multiple Choice Questions)** | | | | | |
| **Q.**  **No** | **Questions** | | **Course Outcome** | | **Competence**  **BT Level** |
| **1** | | **What kind of library is Standard Template Library?** | | --- | | a) Polymorphic | | b) Generic | | c) Both Polymorphic & Generic | | d) None of the mentioned  Ans:b | | CO6 | | BT1 | |
| **2** | **To what type of object does the container can be instantiated?**   1. int 2. float 3. double 4. any type of object   Ans:d | CO6 | | BT1 | |
| **3** | **What type of class template is list?**   1. Class-based 2. Node-based 3. Method-based 4. None of the mentioned   Ans:b | CO6 | | BT2 | |
| **4** | **What type of access does deque and vector provide?**   1. Linear access 2. Parallel access 3. Random access 4. None of the mentioned   Ans:c | CO6 | | BT2 | |
| **5** | **Where does the vector add the item?**   1. End 2. Insert 3. Middle 4. None of the mentioned   Ans:a | CO6 | | BT1 | |
| **6** | **Which are not full container classes in C++?**   1. Sequence container 2. Associative container 3. Container adaptor 4. None of the mentioned   Ans:c | CO6 | | BT1 | |
| **7** | **What is the lifetime of the element in container?**   1. Whole program 2. Outside the block 3. Everywhere 4. Only on that container   Ans:d | CO6 | | BT1 | |
| **8** | **Which operator is used to insert the data into file?**   1. >> 2. << 3. < 4. None of the Above   Ans: b | CO6 | | BT1 | |
| **9** | **Which function is used to position back from the end of file object?**   1. seekg 2. seekp 3. both seekg&seekp 4. None of the Above   Ans:a | CO6 | | BT1 | |
| **10** | **How many objects are used for input and output to a string?**   1. 1 2. 2 3. 3 4. 4   Ans:c | CO6 | | BT2 | |
| **11** | **Which is used to handle the exceptions in c++?**   1. catch handler 2. handler 3. exception handler 4. None of the Mentioned   Ans:c | CO6 | | BT1 | |
| **12** | **Which type of program is recommended to include in try block?**   1. static memory allocation 2. dynamic memory allocation 3. const reference 4. pointer   Ans:b | CO6 | | BT1 | |
| **13** | **Which statement is used to catch all types of exceptions?**   1. catch() 2. catch(Test t) 3. catch(…) 4. None of the Above   Ans:c | CO6 | | BT1 | |
| **14** | **What do Associate containers implement?**   1. Arrays 2. Associative arrays 3. Functional Arrays 4. Static arrays    Ans: b | CO6 | | BT1 | |
| **15** | By using which of the following the elements in the associate container can be efficiently accessed?  1. Key 2. Position 3. Both Key & Position 4. Value    Ans: a | CO6 | | BT3 | |
| **16** | **How many items are presented in the associate container?**   1. 2 2. 3 3. 4 4. 5    Ans: c | CO6 | | BT1 | |
| **17** | **What are the containers?**   1. Containers store objects and data 2. Containers stores all the algorithms 3. Containers contain overloaded functions 4. Containers contain set of Iterators   Ans: a | CO6 | | BT1 | |
| **18** | **In how many categories, containers are divided?**   1. 1 2. 2 3. 3 4. **4** | CO6 | | BT2 | |
| **19** | **What are the Sequence Containers?**   1. Containers that implements data structures which can be accessed sequentially 2. Containers that implements sorted data structures for fast search in O(logn) 3. Containers that implements unsorted(hashed) data structures for quick search in O(1) 4. Containers that implements data structures which can be accessed non-sequentially   Ans:a | CO6 | | BT2 | |
| **20** | **How many Sequence Containers are provided by C++?**   1. 2 2. 3 3. 4 4. 5   Ans:d | CO6 | | BT1 | |
| **21** | **What is the Standard Template Library?** a) Set of C++ template classes to provide common programming data structures and functions b) Set of C++ classes c) Set of Template functions used for easy data structures implementation d) Set of Template data structures only  Ans:a | CO6 | | BT1 | |
| **22** | **What are Unordered Associative Containers?** a) Containers that implements data structures which can be accessed sequentially b) Containers that implements sorted data structures for fast search in O(logn) c) Containers that implements unsorted(hashed) data structures for quick search in O(1) d) Containers that implements data structures which can be accessed non-sequentially  Ans:c | CO6 | | BT1 | |
| **23** | **What are Iterators?** a) Iterators are used to iterate over C-like arrays b) Iterators are used to iterate over pointers c) Iterators are used to point memory addresses of STL containers d) Iterators are used to iterate over functions  **Ans:** C | CO6 | | BT1 | |
| **24** | **Which header file is used for Iterators?** a) <iterator> b) <algorithm> c) <iter> d) <loopIter>  **Ans:**a | CO6 | | BT1 | |
| **25** | Consider that the variable str is of type std:string. What is the correct way to get the C-style string from str?   | **A.** Cast str to const char\* as in((const char\*)&str) | | --- | | **B.** Use str.get\_c\_style\_string() | | **C.** Use str.c\_str() | | **D.** Use str.data()  Ans:C | | CO6 | | BT2 | |
| **PART B (4 Marks)** | | | | | |
| **1** | What are containers in C++ STL? | CO6 | | BT1 | |
| **2** | What are the 3 entities of STL in C++? | CO6 | | BT1 | |
| **3** | What is true about his statement in C++?  std::vector<int> vecInts(5); | CO6 | | BT2 | |
| **4** | Justify your answers Is it possible to initialize any Vector with an Array in C++? | CO6 | | BT1 | |
| **5** | Difference between [Vector Vs List](https://thispointer.com/difference-between-vector-and-list-in-c/) | CO6 | | BT2 | |
| **6** | [Different Ways to Initialize a List](https://thispointer.com/different-ways-to-initialize-a-list-in-c/) and elaborate them | CO6 | | BT2 | |
| **7** | How to [erase elements from List using Iterators](https://thispointer.com/how-to-erase-elements-from-a-list-in-c-using-iterators/) | CO6 | | BT1 | |
| **8** | How to [Remove Elements from a List while Iterating](https://thispointer.com/how-to-remove-elements-from-a-list-while-iterating/) | CO6 | | BT1 | |
| **PART C (12 Marks)** | | | | | |
| **1** | Give syntax of and explain various functions related to ifstream and ofstream classes: seekp(), getline(),hide(),tail(). | CO6 | | BT3 | |
| **2** | Explain the use of ifstream and ofstream classes for file input and output. | CO6 | | BT3 | |
| **3** | Explain the file operation functions in C++ to manipulate the position of file pointers in a random access file. | CO6 | | BT3 | |
| **4** | What is the purpose of push\_back(), push\_front(), pop\_back() and pop\_front() functions of a list. | CO6 | | BT2 | |
| **5** | What does this function do?  void func() {  std::vector<std::string> vecOfString(5, "Hi");  for (std::string str : vecOfStr)  std::cout << str << std::endl;  } | CO6 | | BT2 | |

**Note:**

1. **BT Level –** Blooms Taxonomy Level
2. **CO – Course Outcomes**

BT1 –Remember BT2 – Understand BT3 – Apply BT4 – Analyze BT5 – Evaluate BT6 – Create