|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Faculty: Mr. Hule Kuldeep A.**  **Name of Subject: Object Oriented Programming (210243)**  **Year and Branch: SE(COMP)** | | | | | | |
| **Q. No.** | **Description Question** | **Choices** | **Unit No.** | **Difficulty levels (Easy/Medium/Hard)** | **BT Levels** | **Correct Answer** |
| **Unit I- Fundamentals of Object-Oriented Programming** | | | | | | |
| 1 | What is the difference between struct and class in terms of Access Modifier? | 1. By default, all the struct members are private while by default class members are public. 2. By default, all the struct members are protected while by default class members are private. 3. By default, all the struct members are public while by default class members are private. 4. By default, all the struct members are public while by default class members are protected. | 1 | Medium | 4 | **C** |
| 2 | The default access level assigned to members of a class is \_\_\_\_\_\_ | 1. Private 2. Public 3. Protected 4. Needs to be assigned | 1 | Easy | 1 | **A** |
| 3 | Which of the following operators allow defining the member functions of a class outside the class? | 1. :: 2. ? 3. :? 4. % | 1 | Medium | 2 | **A** |
| 4 | Which one of the following is not a fundamental data type in C++? | 1. Float 2. string 3. Int 4. Char | 1 | Easy | 1 | B |
| 5 | If a member needs to have unique value for all the objects of that same class, declare the member as\_\_\_\_\_\_ | 1. Global variable outside class 2. Local variable inside constructor 3. Static variable inside class 4. Dynamic variable inside class | 1 | Hard | 2 | C |
| 6 | What is a constructor? | 1. A class automatically called whenever a new object of this class is created. 2. A class automatically called whenever a new object of this class is destroyed. 3. A function automatically called whenever a new object of this class is created. 4. A function automatically called whenever a new object of this class is destroyed. | 1 | Medium | 2 | A |
| 7 | #include<iostream>  using namespace std;  class X {  public:      int x;  };  int main() {      X a = {10};      X b = a;      cout <<a.x <<" " <<b.x;      return 0;  } | 1. Compiler Error 2. 10 followed by Garbage Value 3. 10 10 4. 10 0 | 1 | Medium | 3 | C |
| 8 | Which of the following interface determines how your program will be used by other program? | 1. Public 2. Private 3. Protected 4. None of these | 1 | Easy | 1 | A |
| 9 | Under what conditions a destructor destroys an object? | 1. Scope of existence has finished 2. Object dynamically assigned and it is released using the operator delete. 3. Program terminated. 4. Both a and b. | 1 | Hard | 2 | A,B |
| 10 | If a member needs to have unique value for all the objects of that same class, declare the member as | 1. Global variable outside class 2. Local variable inside constructor 3. Static variable inside class 4. Dynamic variable inside class | 1 | Medium | 2 | A |
| 11 | Which one of the following is not a valid reserved keyword in C++? | 1. explicit 2. public 3. implicit 4. private | 1 | Easy | 1 | C |
| 12 | Variables declared in the body of a particular member function are known as data members and can be used in all member functions of the class. | 1. TRUE 2. FALSE | 1 | Easy | 2 | B |
| 13 | In a class definition, data or functions designated private are accessible | 1. to any function in the program. 2. only if you know the password. 3. to member functions of that class. 4. only to public members of the class. | 1 | Medium | 2 | C |
| 14 | In an assignment statement, the value on the left of the equal sign is always equal to the value on the right. | 1. TRUE 2. FALSE | 1 | Easy | 1 | B |
| 15 | It’s perfectly all right to use variables of different data types in the same arithmetic expression. | 1. TRUE 2. FALSE | 1 | Medium | 2 | A |
| 16 | For the object for which it was called, a const member function | 1. can modify both const and non-const member data. 2. can modify only const member data. 3. can modify only non-const member data. 4. can modify neither const nor non-const member data. | 1 | Hard | 2 | D |
| 17 | Dividing a program into functions | 1. is the key to object-oriented programming. 2. makes the program easier to conceptualize. 3. may reduce the size of the program. 4. Option B and C | 1 | Medium | 3 | D |
| 18 | A function argument is | 1. a variable in the function that receives a value from the calling program. 2. a way that functions resist accepting the calling program’s values. 3. a value sent to the function by the calling program. 4. a value returned by the function to the calling program. | 1 | Easy | 3 | C |
| 19 | Which of the following can legitimately be passed to a function? | 1. A constant 2. A variable 3. A structure 4. All of the above | 1 | Easy | 4 | A,B,C |
| 20 | When arguments are passed by value, the function works with the original arguments in the calling program. | 1. TRUE 2. FALSE | 1 | Easy | 2 | B |
| 21 | When a function returns a value, the entire function call can appear on the right side of the equal sign and be assigned to another variable. | 1. TRUE 2. FALSE | 1 | Easy | 1 | A |
| 22 |  | 1. 10 30 50 2. 60 50 40 3. 60 40 20 4. 60 60 60 | 1 | Medium | 3 | D |
| 23 |  | 1. 8 2. 12 3. 16 4. 24 | 1 | Hard | 3 | C |
| 24 | Which of the following feature is not supported by C++? | 1. Exception Handling 2. Reflection 3. Operator Overloading 4. Namespace | 1 | Easy | 2 | B |
| 25 | What will be the output of the following program? | 1. Compilation Error: display() cannot be accessed in application 2. Compilation Error:Test class object cannot be accessed in function Demo 3. Compilation Error: Variable x is private in Test 4. Both A and B | 1 | Easy | 2 | C |
| 26 | What value will be printed for data.i\_? | 1. 10 220.5 230.5 Unpredictable Value 2. 220 3. 230.5 4. Unpredictable Value | 1 | Medium | 2 | D |
| 27 | When an array name is passed to a function, the function | 1. accesses exactly the same array as the calling program. 2. refers to the array using a different name than that used by the calling program. 3. refers to the array using the same name as that used by the calling program. 4. A and B | 1 | Medium | 1 | D |
| 28 | Suppose **a and b** are integer variables and we form the sum a + b. Now suppose **c and d** are floating-point variables and we form the sum c + d. The two +operators here are clearly being used for different purposes. This is an example of \_\_\_\_\_\_\_\_\_\_\_\_ | 1. Operator Overloading 2. Inheritance 3. Function Overloading 4. Constructor | 1 | Easy | 2 | A |
| 29 | C++ programmers concentrate on creating, which contain data members and the member functions that manipulate those data members and provide services to clients. | 1. Structures 2. Classes 3. Objects 4. Function | 1 | Easy | 1 | B |
| 30 |  | 1. 0 0 2. 5 6 3. Compilation error: no default constructor 4. Compilation error: constructor is private | 1 | Medium | 2 | D |
| 31 | Which of the following is FALSE about references in C++ | 1. A reference must be initialized when declared 2. Once a reference is created, it cannot be later made to reference another object; it cannot be reset 3. References cannot be NULL 4. References cannot refer to constant value | 1 | Medium | 3 | D |
| 32 | Which of the following is true about constructors?  1 They cannot be virtual.  2 They cannot be private.  3 They are automatically called by new operator | 1. All 1, 2, and 3 2. Only 1 and 3 3. Only 1 and 2 4. Only 2 and 3 | 1 | Medium | 3 | B |
| 33 |  | 1. \*lname, \*fname, \*mname 2. \*mname, \*lname, \*fname 3. \*fname, \*lname, \*mname 4. \*lname, \*mname, \*fname | 1 | Hard | 6 | C |
| 34 | Which of the following is true about inline functions and macros? | 1. Inline functions do type checking for parameters, macros don't 2. Macros cannot have return statement, inline functions can 3. Macros are processed by pre-processor and inline functions are processed in later stages of compilation. 4. All of the above | 1 | Medium | 2 | D |
| 35 | In C++, const qualifier can be applied to  1 Member functions of a class  2 Function arguments  3 To a class data member which is declared as static  4 Reference variables | 1. Only 1, 2 and 3 2. Only 1, 2 and 4 3. All 4. Only 1, 3 and 4 | 1 | Medium | 2 | C |
| 36 |  | 1. 111222 2. 1112 3. 1212 4. 1222 | 1 | Hard | 3 | B |
| 37 |  | 1. fun(10) invokes Function-1 and fun(10,20) invokes Function-2 2. fun(10) invokes Function-3 and fun(10,20) invokes Function-2 3. fun(10) invokes Function-1 and fun(10,20) is an ambiguous call 4. Both function calls are ambiguous | 1 | Hard | 3 | D |
| 38 | Objects are destroyed in the reverse order of its creation | 1. True 2. False | 1 | Easy | 2 | A |
| 39 | What is the output of the above program? | 1. 5 5 5 2. 0 0 0 3. 0 5 0 4. 5 0 5 | 1 | Easy | 2 | C |
| 40 | Which of the following concepts of OOPS means exposing only necessary information to client? | 1. Encapsulation 2. Abstraction 3. Data hiding 4. Data binding | 1 | Easy | 1 | B |
| 41 |  | 1. S1 Created S2 Created S1 Destroyed S2 Destroyed 2. S1 Created S2 Created S2 Destroyed S1 Destroyed 3. S2 Created S1 Created S2 Destroyed S1 Destroyed 4. S1 Created S2 S1 Destroyed Created S2 Destroyed | 1 | Medium | 3 | A |
| 42 |  | 1. Square=4 Rectangle=5 Trapezium=10 Destroyed:3 Destroyed:2 Destroyed:1 2. Square=4 Rectangle=6 Trapezium=10 Destroyed:1 Destroyed:2 Destroyed:3 3. Trapezium=10 Rectangle=6 Square=4 Destroyed:1 Destroyed:2 Destroyed:3 4. Trapezium=10 Rectangle=6 Square=4 Destroyed:3 Destroyed:2 Destroyed:1 | 1 | Hard | 4 | A |
| 43 |  | 1. friend void display(const myClass&) 2. void friend display(const myClass&) 3. void display(const myClass&) 4. friend display(const myClass&) | 1 | Hard | 3 | A |
| 44 |  | 1. 5 2. 0 3. <Unpredicted value> 4. Error: C++ forbids in-class initialization of non-const static member | 1 | Hard | 2 | D |
| 45 | .... constructor is used for copying the object of same class type. | 1. Copy 2. Default 3. Parameterized 4. None of above | 1 | Easy | 2 | A |
| 46 |  | 1. 0 1 2 2. 1 0 2 3. 0 2 1 4. 1 2 0 | 1 | Medium | 2 | A |
| 47 |  | 1. void static update(int a) 2. void update(int a) 3. friend void update(int a) 4. void static friend update(int a) | 1 | Hard | 3 | D |
| 48 | Default return type of C++ main( ) is ..... | 1. float 2. void 3. int 4. pointer | 1 | Easy | 1 | C |
| 49 |  | 1. using namespace e; 2. using namespace e::x; 3. using e::x; 4. using namespace ::x; | 1 | Medium | 2 | A |
| 50 |  | 1. \_a,\_b,\_c 2. \_c,\_b,\_a 3. a,b,c 4. c,b,a | 1 | Medium | 2 | B |
| 51 | Objects are to classes as variables are to data types. | 1. True 2. False | 1 | Easy | 2 | A |
| 52 | A const member function prevents modification of any of its class’s member data. | 1. True 2. False | 1 | Easy | 2 | A |
| 53 | When an object is passed to function by value or returned from a function by value the copy constructor gets called. | 1. True 2. False | 1 | Medium | 1 | A |
| 54 | The compiler always provides a zero-argument constructor by default. | 1. True 2. False | 1 | Medium | 1 | A |
| 55 | Static memory allocation takes place during compilation, whereas dynamic memory allocation takes place during executeion | 1. True 2. False | 1 | Easy | 1 | A |
| 56 | If member functions of a class are defined outside the class, it is not necessary to declare them inside the class. | 1. True 2. False | 1 | Medium | 2 | A |
| 57 | x != y is the same as (x > y || x < y) | 1. True 2. False | 1 | Medium | 2 | A |
| 58. | A variable defined in an inner block may not have the same name as a variable defined in the outer block. | 1. True 2. False | 1 | Easy | 1 | A |
| 59 | C++ allows you to create arrays with three or more dimensions. | 1. True 2. False | 1 | Easy | 1 | A |
| 60 | When an array is passed to a function, the function has access to the original array. | 1. True 2. False | 1 | Medium | 1 | A |
| 61 | A static data member’s definition appears in the class declaration, but the variable is actually declared outside the class | 1. True 2. False | 1 | Easy | 1 | F |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Faculty: Mr. Hule Kuldeep A.**  **Name of Subject: Object Oriented Programming (210243)**  **Year and Branch: SE(COMP)** | | | | | | |
| **Q. No.** | **Description Question** | **Choices** | **Unit No.** | **Difficulty levels (Easy/Medium/Hard)** | **BT Levels** | **Correct Answer** |
| **Unit - II Inheritance and Pointers** | | | | | | |
| 1 | Choose the right option  string\* x, y; | 1. x is a pointer to a string, y is a string 2. y is a pointer to a string, x is a string 3. both x and y are pointer to string types 4. none of the mentioned | 2 | Easy | 1 | A |
| 2 | Which of the following is illegal? | 1. int \*ip; 2. string s, \*sp = 0; 3. int i; double\* dp = &i; 4. int \*pi = 0; | 2 | Easy | 1 | C |
| 3 | #include <iostream>  using namespace std;  int main() {  int a = 5, b = 10, c = 15;  int \*arr[ ] = {&a, &b, &c};  cout <<arr[1];  return 0;  }  What is the output of the above program | 1. 10 2. 15 3. 20 4. Random number | 2 | Medium | 2 | D |
| 4 |  | 1. ABCDEFGHIJ 2. AAAAAAAAA 3. JJJJJJJJJJJJ 4. None | 2 | Medium | 3 | A |
| 5 | What we will not do with function pointers? | 1. allocation of memory 2. de-allocation of memory 3. both a & b 4. None | 2 | Easy | 2 | C |
| 6 |  | 1. 12 2. 8 3. 4 4. 1 | 2 | Easy | 1 | B |
| 7 |  | 1. base.f1 2. derived.f1 3. base.f1   derived.f1   1. compilation error at LINE-1: no matching function for call derived.f1() | 2 | Medium | 2 | D |
| 8 |  | 1. Class A   Class B   1. Class A   Class A   1. Class B   Class A   1. Class B   Class B | 2 | Hard | 3 | B |
| 9 |  | 1. Hello   Hello   1. Hello   Hi   1. Hi   Hello   1. Compilation error at LINE1: argument mismatch | 2 | Easy | 2 | A |
| 10 |  | 1. LINE-1 2. LINE-2 3. Both LINE-1 and LINE-2 4. No Compilation error | 2 | Medium | 2 | C |
| 11 |  | 1. A B C ~C ~B ~A 2. A C ~C ~A 3. A A B C ~C ~B ~A ~A 4. A A B C ~A ~A ~B ~C | 2 | Easy | 2 | B |
| 12 |  | 1. print(); or A::print() 2. A::print; 3. A.print(); 4. None of these | 2 | Easy | 2 | B |
| 13 | Derived class\_\_\_ cannot access from base class | 1. Constructor 2. Destructors 3. copy constructor 4. all of the above | 2 | Easy | 2 | D |
| 14 | We have to define a constructor for the derived class must be required\_\_\_\_. | 1. if base class constructor does not require arguments 2. if base/parent class constructor required arguments 3. no need 4. always | 2 | Easy | 2 | D |
| 15 | Inheritance can be done using :: symbol | 1. True 2. False | 2 | Easy | 2 | B |
| 16 | Adding a derived class to a base class requires fundamental changes to the base class. | 1. True 2. False | 2 | Easy | 2 | B |
| 17 |  | 1. A::0 B::1 C::1 A::5 B::6 C::6   ~C ~B ~A ~C ~B ~A   1. A::0 B::1 C::2 ~C ~B ~A   A::5 B::6 C::7 ~C ~B ~A   1. A::0 B::1 C::1 ~C ~B ~A   A::5 B::6 C::6 ~C ~B ~A   1. A::0 B::1 C::2 A::5 B::6 C::7   ~C ~B ~A ~C ~B ~A | 2 | Hard | 3 | A |
| 18 | Make a correct sequence of a statement  i)destructor of derived class is called  ii)destructor of base class is called  iii)constructor of derived class is called  iv)constructor of base class is called | 1. i,ii,iv,iii 2. iv,iii,ii,i 3. iv,iii,i,ii 4. i,ii,iii,iv | 2 | Medium | 2 | C |
| 19 | Virtual base class is used to \_\_\_\_. | 1. to perform operator overloading 2. to perform function overloading 3. to remove ambiguity in multiple inheritance 4. all of the above | 2 | Easy | 3 | C |
| 20 |  | 1. 10 5 2. 5 10 3. Compilation error at LINE-1 4. Compilation error at LINE-2 | 2 | Medium | 2 | D |
| 21 |  | 1. \*p 2. \*\*p 3. (\*p)[3] 4. \*p[3] | 2 | Medium | 2 | C |
| 22 |  | 1. 5 2. 20 3. 15 4. Unpredictable value | 2 | Hard | 3 | B |
| 23 |  | 1. STMT-1 2. STMT-2 3. STMT-3 4. STMT-4 | 2 | Medium | 3 | B |
| 24 |  | 1. const int \*a=new int(2){5,10}; 2. const int \*a=new int[2]{5,10}; 3. const int \*a=new int[2](5,10); 4. const int \*a=new int(2)(5,10); | 2 | Medium | 2 | B |
| 25 | Pick up the correct statement | 1. protected member from base class can be accessed by own class and its all subclasses 2. protected member are not inherited by any other class 3. Protected member are combination of public and private access member 4. all of the above | 2 | Medium | 2 | D |
| 26 | A pointer that is pointing to NOTHING is called \_\_\_\_ | 1. VOID Pointer 2. DANGLING Pointer 3. NULL Pointer 4. WILD Pointer | 2 | Easy | 2 | C |
| 27 | A pointer can be initialized with | 1. Null 2. Zero 3. Address of an object of same type 4. All of the above | 2 | Easy | 1 | D |
| 28 | A void pointer cannot point to which of these? | 1. Methods in c++ 2. Class member in c++ 3. Both A & B 4. None of the above | 2 | Medium | 2 | D |
| 29 | Which is the pointer which denotes the object calling the member function? | 1. variable pointer 2. this pointer 3. Null pointer 4. Zero pointer | 2 | Medium | 4 | B |
| 30 |  | 1. 24 2. 35 3. 16 4. 36 | 2 | Medium | 2 | D |
| 31 | What will be the output of the following program? | 1. 32, A 2. 32, a 3. 129, a 4. 129, A | 2 | Medium | 2 | C |
| 32 | What will be the output of the following program? | 1. 20 2. 21 3. 15 4. Compile error | 2 | Medium | 2 | D |
| 33 | What will be the output of the following program? | 1. 12 2. 15 3. 14 4. error | 2 | Medium | 2 | C |
| 34 | What is the output of the following C++ program? | 1. 1 2 3 4 5 2. Address of the elements 3. run time error 4. None of the mentioned | 2 | Medium | 2 | D |
| 35 | Assume a class Derv that is privately derived from class Base. An object of class Derv located in main() can access | 1. public members of Derv. 2. protected members of Derv. 3. private members of Derv. 4. public members of Base. | 2 | Easy | 1 | D |
| 36 | A class hierarchy | 1. shows the same relationships as an organization chart. 2. describes “has a” relationships. 3. describes “is a kind of” relationships. 4. shows the same relationships as a family tree. | 2 | Easy | 2 | C |
| 37 | True or False: A class **D** can be derived from a class **C**, which is derived from a class **B**, which is derived from a class **A**. | 1. True 2. False | 2 | Easy | 2 | A |
| 38 | When deriving a class with public inheritance, public members of the base class become \_\_\_\_\_\_\_\_\_\_ members of the derived class, and protected members of the base class become \_\_\_\_\_\_\_\_\_\_\_members of the derived class. | 1. private, private 2. public, protected. 3. protected, protected. 4. private, protected | 2 | Medium | 2 | B |
| 39 | Which of the following is true about this pointer? | 1. It is passed as a hidden argument to all function calls 2. It is passed as a hidden argument to all non-static function calls 3. It is passed as a hidden argument to all static functions 4. None of the above | 2 | Medium | 3 | B |
| 40 | What will be the output of the program? | 1. d9   9   1. d9d9 2. d9 3. Compile time error | 2 | Medium | 3 | D |
| 41 | What is meaning of following declaration?  int(\*ptr[5])(); | 1. ptr is pointer to function. 2. ptr is array of pointer to function. 3. ptr is pointer to such function which return type is array. 4. ptr is pointer to array of function. | 2 | Hard | 1 | B |
| 42 | Identify the correct sentence regarding inequality between reference and pointer. | 1. we cannot create the array of reference. 2. we can create the Array of reference. 3. we can use reference to reference. 4. None of these | 2 | Easy | 3 | A |
| 43 | What will be the output of the above program? | 1. address address value 2. address value value 3. address address address 4. compilation error | 2 | Medium | 3 | B |
| 44 | What will be the output of the above program? | 1. 10, 20, 30, 40 2. Striker>10 Next@50 Last@40 Reset To0 3. Striker>10 Next@40 Last@50 Reset To0 4. Striker> Next@ Last@ Reset To | 2 | Easy | 3 | B |
| 45 | Which of the following statements is correct? | 1. Base class pointer cannot point to derived class. 2. Derived class pointer cannot point to base class. 3. Pointer to derived class cannot be created. 4. Pointer to base class cannot be created. | 2 | Hard | 4 | B |
| 46 | What is the output of the following program? | 1. 11 2. 10 3. 01 4. 00 | 2 | Hard | 2 | B |
| 47 | What is the output of the following program?    2 | 1. 8 2. 16 3. 128 4. 64 | 2 | Hard | 3 | C |
| 48 | The correct statement for a function that takes pointer to a float, a pointer to a pointer to a char &return a pointer to a integer is | 1. int\*\*fun(float\*\*, char\*\*) 2. int \*fun(float\*, char\*) 3. int \*\*\*fun(float\*, char\*\*) 4. int \*\*\*fun(\*float, \*\*char) | 2 | Easy | 3 | A |
| 49 | The new not only allocates memory but also calls the object’s constructor. | 1. True 2. False | 2 | Easy | 3 | A |
| 50 | Deleting a NULL pointer is sage & is guaranteed to do nothing. | 1. True 2. False | 2 | Easy | 4 | A |
| 51 | In C++ to reallocate memory we should use the renew operator. | 1. True 2. False | 2 | Easy | 2 | B |
| 52 | The new operator always returns a pointer of appropriate type, whereas, malloc() returns a void pointer which needs to be typecast explicitly. | 1. True 2. False | 2 | Medium | 2 | A |
| 53 | The this pointer can be used in a class specific overloaded new operator function. | 1. True 2. False | 2 | Easy | 2 | A |
| 54 | The way a derived class member function can access base class protected & public members, the base class member functions can access protected & public member functions of derived class | 1. True 2. False | 2 | Medium | 1 | B |
| 55 | There is no difference between private & protected inheritance | 1. True 2. False | 2 | Easy | 1 | B |
| 56 | In public inheritance the protected members of the base class become public for the functions outside the derived class. | 1. True 2. False | 2 | Easy | 1 | B |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Faculty: Mr. Hule Kuldeep A.**  **Name of Subject: Object Oriented Programming (210243)**  **Year and Branch: SE(COMP)** | | | | | | |
| **Q. No.** | **Description Question** | **Choices** | **Unit No** | **Difficulty levels (Easy/Medium/Hard)** | **BT Levels** | **Correct Answer** |
| **Unit- III Polymorphism** | | | | | | |
| 1 | Which of the following in Object Oriented Programming is supported by Function overloading and default arguments features of C++ | 1. Inheritance 2. Polymorphism 3. encapsulation 4. None | 3 | Easy | 1 | B |
| 2 | Which function cannot be overloaded in C++ program? | 1. Virtual function 2. member function 3. Static function 4. All can be overloaded | 3 | Easy | 2 | C |
| 3 | Find out the output:  #include<iostream>  using namespace std;  int fun(int x = 0, int y = 0, int z)  {  return (x + y + z);  }  int main()  {  cout << fun(10);  return 0;  } | 1. 10 2. 0 3. 20 4. Compiler error | 3 | Easy | 2 | D |
| 4 | Function overloading is run time polymorphisms | 1. True 2. False | 3 | Easy | 1 |  |
| 5 | Operator overloading is also called one form of polymorphism because\_\_\_\_\_\_\_. | 1. the overloaded operators have many forms 2. the overloaded operators can be declared virtual 3. the overloaded function can perform various tasks depending upon the type of object 4. None of these | 3 | Easy | 1 | A |
| 6 | Overloading means | 1. two or more methods in the same class that have same name 2. calling the method which has actual parameters 3. two or more methods having same name but present in different class 4. none of these | 3 | Easy | 1 | A |
| 7 | Polymorphism is supported by the c++ by using following ways | 1. function overloading 2. operator overloading 3. virtual functions 4. all of the above | 3 | Medium | 2 | D |
| 8 | Selecting the appropriate overloaded function by the compiler is known as | 1. late binding 2. early binding 3. both a and b 4. none of the above | 3 | Medium | 2 | D |
| 9 | Which of the following operator cannot be overloaded | 1. scope resolution operator(::) 2. Size of operator (sizeof[]) 3. Conditional operator(?:) 4. All of the above | 3 | Easy | 2 | D |
| 10 | We are overloading a unary operator without friend function how many argument we have to pass | 1. 0 2. 1 3. 2 4. 3 | 3 | Easy | 1 | A |
| 11 | What is true about the operator overloading | 1. with friend function we need to pass two arguments for binary operator 2. with friend function we need to pass one arguments for unary operator 3. both a and b 4. none of the above | 3 | Medium | 2 | C |
| 12 | Converting from small to larger data type is known as \_\_\_\_. | 1. Promotion 2. Operator 3. Polymorphism 4. none of the above | 3 | Easy | 2 | A |
| 13 | pick the correct option | 1. We can make the instance of the abstract class 2. We cannot make the instance of the abstract class 3. both a and b 4. none of the above | 3 | Medium | 2 | B |
| 14 | When we have to use the mutable keyword | 1. data member to change within a const member function 2. not allow the data member to change within a const member function 3. it will copy the values of the variable 4. none of the above mentioned | 3 | Medium | 4 | A |
| 15 | Assume a class C with objects obj1, obj2, and obj3. For the statement obj3 = obj1 - obj2 to work correctly, the overloaded - operator must | 1. take two arguments. 2. return a value. 3. use the object of which it is a member as an operand. 4. Both B and C | 3 | Medium | 4 | D |
| 16 | To convert from a user-defined class to a basic type, you would most likely use | 1. a built-in conversion operator. 2. a one-argument constructor. 3. an overloaded = operator. 4. a conversion operator that’s a member of the class. | 3 | Medium | 2 | D |
| 17 | An overloaded operator always requires one less argument than its number of operands. | 1. True 2. False | 3 | Easy | 3 | A |
| 18 | Complete operator overloading for structure at Line-1 so that the output is “5+i3” | 1. complex operator+(complex &c1, complex &c2) 2. complex operator+(const complex &c1, const complex &c2) 3. operator+(complex &c1, complex &c2) 4. complex +(complex &c1, complex &c2) 5. both a &b | 3 | Medium | 2 | E |
| 19 | This program will give error without LINE-1. Fill in the blank at LINE-1 to avoid any compilation error | 1. friend void display(const myClass&) 2. void friend display(const myClass&) 3. void display(const myClass&) 4. friend display(const myClass&) 5. both a and b | 3 | Medium | 2 | A |
| 20 |  | 1. 5 5 2. 6 6 3. 6 5 4. 5 6 | 3 | Medium | 3 | C |
| 21 | Which of the following operators are overloaded by default by the compiler?  1) Comparison Operator ( = = )  2) Assignment Operator ( = ) | 1. Both 1 and 2 2. Only 1 3. Only 2 4. None of the two | 3 | Medium | 2 | C |
| 22 | Pick the incorrect statement from the following | 1. The overloaded operators follow the syntax rules of original operator. 2. Only existing operators can be overloaded 3. Overloaded operator must have at least one operand of its class type 4. Overloaded operators can change the meaning of the original operator | 3 | Medium | 2 | D |
| 23 |  | 1. 2 5   4 6   1. 6 5   2 11   1. 6 11   2 5   1. 2 5   6 11 | 3 | Medium | 2 | A |
| 24 | A pure virtual function is a virtual function that | 1. causes its class to be abstract. 2. returns nothing. 3. is used in a base class. 4. A and C | 3 | Easy | 2 | A |
| 25 | Virtual functions allow you to | 1. create an array of type pointer-to-base class that can hold pointers to derived classes. 2. create functions that can never be accessed. 3. group objects of different classes so they can all be accessed by the same function code. 4. use the same function call to execute member functions of objects from different classes. | 3 | Hard | 4 | D |
| 26 | The operator functions may be invoked \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | 1. Implicitly 2. Explicitly 3. Both a & b 4. None of these | 3 | Easy | 3 | C |
| 27 | The symbol \*\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | 1. Can be overloaded 2. Cannot be overloaded as it is not a C++ operator 3. Cannot be overloaded, as on overloading its meaning shall be changed 4. None of these | 3 | Medium | 3 | B |
| 28 | What is the return type of the conversion operator? | 1. void 2. int 3. float 4. no return type | 3 | Medium | 3 | D |
| 29 | What will be the output of the following C++ code?  #include <iostream>  using namespace std;  int main() {  double a = 21.09399;  float b = 10.20; int c ;  c = (int) a;  cout << c ;  c = (int) b;  cout << c ;  return 0;  } | 1. 2110 2. 1210 3. 21 4. 121 | 3 | Easy | 2 | A |
| 30 | Use of virtual functions implies | 1. Dynamic Binding 2. Overloading 3. Static Binding 4. None of these | 3 | Easy | 2 | A |
| 31 | Pointer to the base class can hold address of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | 1. Only Base class object 2. Only Derived class object 3. Base class object as well as derived class object 4. None of these | 3 | Easy | 2 | C |
| 32 | Which of the following is true about pure virtual functions?  1) Their implementation is not provided in a class where they are declared.  2) If a class has a pure virtual function, then the class becomes abstract class and an instance of this class cannot be created. | 1. Both 1 and 2 2. Only 1 3. Only 2 4. Neither 1 nor 2 | 3 | Easy | 3 | A |
| 33 | Can a destructor be virtual? Will the following program compile?  #include <iostream>  using namespace std;  class Base {  public:  virtual ~Base() {}  };  int main() {  return 0;  } | 1. Yes 2. No 3. Can’t Say | 3 | Medium | 2 | A |
| 34 | Explain what happen in following program  #include<iostream> using namespace std;  class Base{  public: virtual void show()=0;  };  int main()  {  Base b, \*bp;  return 0;  } | 1. There are compiler errors in line Base b & Base bp 2. There is compiler error in line Base b; 3. There is compiler error in line Base \*bp; 4. No compiler error | 3 | Hard | 4 | B |
| 35 | Choose the correct the statement from following  i)inheritance supported in terms of single, multiple, multilevel, hybrid inheritance  ii) polymorphisms is supported by function overloading, operator overloading and virtual function  iii) abstraction is not supported by c++ | 1. i is correct 2. only ii is correct 3. i and ii is correct 4. none of the above | 3 | Hard | 3 | C |
| 36 | Which of the following is correct? | 1. C++ allows static type checking 2. C++ allows dynamic type checking. 3. C++ allows static member function to be of type const. 4. C++ allows both static and dynamic type checking | 3 | Easy | 2 | B |
| 37 | Which of the following statement is true?  I) In Procedural programming languages, all function calls are resolved at compile-time  II) In Object Oriented programming languages, all function calls are resolved at compile-time | 1. I only 2. II only 3. Both I and II 4. Neither I nor II | 3 | Medium | 2 | A |
| 38 | How “Late Binding” is implemented in C++? | 1. Using C++ tables 2. Using Indexed virtual tables 3. Using Polymorphic tables 4. Using Virtual tables | 3 | Medium | 2 | D |
| 39 | The operator << when overloaded in a clas | 1. Must be a member of function 2. Must be a non-member function 3. Can be both a and b above 4. Cannot be overloaded | 3 | Easy | 3 | A |
| 40 | The no of arguments required when overloading through friend function is | 1. 0 2. 1 3. 2 4. 3 | 3 | Easy | 2 | B |
| 41 |  | 1. A::fun1   B::fun2   1. A::fun1   A::fun2   1. B::fun1   B::fun2   1. B::fun1   A::fun2 | 3 | Easy | 2 | A |
| 42 |  | 1. LINE-1: Pure virtual function in Base cannot have a body 2. LINE-2: Cannot instantiate abstract class 3. LINE-3: Invalid operator new expression for abstract class type 4. LINE-4: Cannot de-reference a null pointer 5. Both b & c | 3 | Hard | 2 | E |
| 43 |  | 1. base::fun   base::fun   1. base::fun   derived::fun   1. derived::fun   derived::fun   1. derived::fun   base::fun | 3 | Medium | 3 | C |
| 44 |  | 1. 12 8 2. 12 6 3. 10 8 4. 10 6 | 3 | Medium | 1 | D |
| 45 |  | 1. Line-1 2. Line-2 3. Line-3 4. Line-4 | 3 | Medium | 2 | A D |
| 46 |  | 1. A B C ~C ~B ~A 2. A B C ~C ~B 3. A B C ~B ~A 4. A B C ~A | 3 | Medium | 3 | D |
| 47 |  | 1. C::f1   A::f2   1. B::f1   C::f2   1. A::f1   C::f2   1. A::f1   B::f2 | 3 | Medium | 1 | A |
| 48 |  | 1. 0 2. 1 3. 2 4. Garbage | 3 | Medium | 2 | B |
| 49 | Using friend operator function, following perfect set of operators may not be overloaded. | 1. = , ( ) , [ ] , -> 2. <<, = = , [ ] , >> 3. ?, = , ( ) , ++ 4. None of these | 3 | Easy | 2 | A |
| 50 |  | 1. Flower, FlowerWSmell, FlowerWOSmell 2. Flower, FlowerWOSmell, Hibiscus 3. Flower, FlowerWSmell, FlowerWOSmell, Sunflower 4. Flower | 3 | Easy | 1 | B |
| 51 | Virtual functions permit calling of derived class functions using a base class pointer | 1. True 2. False | 3 | Medium | 1 | A |
| 52 | Each object has its own VTABLE. | 1. True 2. False | 3 | Easy | 2 | B |
| 53 | We can access the VTABLE using this pointer | 1. True 2. False | 3 | Easy | 2 |  |
| 54 | Pure virtual function can never have a body | 1. True 2. False | 3 | Medium | 2 | B |
| 55 | Virtual functions permit functions from different classes to be executed through the same function call. | 1. True 2. False | 3 | Easy | 2 | A |
| 56 | Virtual function calls work faster than normal function calls. | 1. True 2. False | 3 | Medium | 2 | A |
| 57 | Virtual destructor ensures a proper calling order for the destructors in the class hierarchy. | 1. True 2. False | 3 | Medium | 2 | A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Faculty: Mr. Hule Kuldeep A.**  **Name of Subject: Object Oriented Programming (210243)**  **Year and Branch: SE(COMP)** | | | | | | |
| **Q. No** | **Description Question** | **Choices** | **Unit No** | **Difficulty levels (Easy/Medium/Hard)** | **BT Levels** | **Correct Answer** |
| **Unit- IV Files and Streams** | | | | | | |
| 1 | Which operator is used to insert the data into file? | 1. >> 2. << 3. < 4. > | 4 | Easy | 1 | B |
| 2 | Which function is used to position back from the end of file object? | 1. seekg 2. seekp 3. both seekg & seekp 4. seekf | 4 | Easy | 1 | A |
| 3 | How many objects are used for input and output to a string? | 1. 1 c. 2 2. 3 d. 4 | 4 | Easy | 2 | C |
| 4 |  | 1. This is sample 2. Sample 3. Error 4. Runtime error | 4 | Hard | 4 | D |
| 5 |  | 1. first 2. second 3. returns first 2 letter or number from the entered word 4. third | 4 | Hard | 4 | C |
| 6 |  | 1. Done 2. Error 3. Runtime error 4. DoneDoneDone | 4 | Hard | 4 | A |
| 7 |  | 1. 100 2. 3.14 3. 314 4. All of the mentioned | 4 | Hard | 4 | D |
| 8 |  | 1. dot operator 2. insertion operator 3. $ symbol 4. @ symbol | 4 | Medium | 2 | A |
| 9 | Which member function is used to determine whether the stream object is currently associated with a file? | 1. is\_open 2. buf 3. string 4. is\_out | 4 | Easy | 1 | A |
| 10 | Which header file is used for reading and writing to a file? | 1. #include<iostream> 2. #include<fstream> 3. #include<file> 4. #include<iomanip> | 4 | Easy | 1 | B |
| 11 | Which stream class is to only write on files? | 1. ofstream 2. ifstream 3. fstream 4. iostream | 4 | Medium | 1 | A |
| 12 | It is not possible to combine two or more file opening mode in open () method. | 1. TRUE 2. FALSE | 4 | Medium | 2 | B |
| 13 | Which of these is the correct statement about eof() ? | 1. Returns true if a file open for reading has reached the next character. 2. Returns true if a file open for reading has reached the next word. 3. Returns true if a file open for reading has reached the end. 4. Returns true if a file open for reading has reached the middle. | 4 | Medium | 2 | C |
| 14 | Which of the following true about FILE \*fp | 1. FILE is a structure and fp is a pointer to the structure of FILE type 2. FILE is a buffered stream 3. FILE is a keyword in C for representing files and fp is a variable of FILE type 4. FILE is a stream | 4 | Medium | 2 | A |
| 15 | Which of the following methods can be used to open a file in file handling? | 1. Using Open ( ) 2. Constructor method 3. Destructor method 4. Both A and B | 4 | Medium | 2 | D |
| 16 | Which function is used for writing data onto file? | 1. getc() 2. getline() 3. write() 4. none of these | 4 | Easy | 1 | C |
| 17 | Which is correct syntax? | 1. myfile:open ("example.bin", ios::out); 2. myfile.open ("example.bin", ios::out); 3. myfile::open ("example.bin", ios::out); 4. myfile.open ("example.bin", ios:out); | 4 | Medium | 2 | B |
| 18 |  | 1. Error 2. Find 3. This is find 4. Runtime error | 4 | Hard | 4 | D |
| 19 |  | 1. A. course 2. Fine 3. Returns fine 2 letter or number from the entered word 4. None of the mentioned | 4 | Hard | 4 | C |
| 20 | ios::trunc is used for ? | 1. If the file is opened for output operations and it already existed, no action is taken. 2. If the file is opened for output operations and it already existed, then a new copy is created. 3. If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one. 4. None of the above | 4 | Medium | 2 | C |
| 21 | By default, all the files are opened in \_\_\_\_\_\_\_\_\_\_\_mode. | 1. Binary 2. Text 3. c. Can’t say | 4 | Easy | 1 | B |
| 22 | It is not possible to combine two or more file opening mode in open () method. | 1. True 2. False | 4 | Easy | 2 | B |
| 23 | Which of the following is not a file opening mode? | 1. ios::ate 2. ios::nocreate 3. ios::noreplace 4. ios::truncate | 4 | Medium | 1 | D |
| 24 | Due to ios::trunc mode, the file is truncated to zero length. | 1. True 2. False | 4 | Easy | 2 | A |
| 25 | If we have object from ofstream class, then default mode of opening the file is \_\_\_\_\_. | 1. ios::in 2. ios::out 3. ios::in|ios::trunc 4. ios::out|ios::trunk | 4 | Medium | 2 | B |
| 26 | \_\_\_\_\_\_\_\_\_\_ is return type of is\_open() function. | 1. int 2. bool 3. float 4. char \* | 4 | Easy | 1 | B |
| 27 | If we have object from fstream class, then what is the default mode of opening the file? | 1. ios::in|ios::out 2. ios::in|ios::out|ios::trunc 3. ios::in|ios::trunc 4. Default mode depends on compiler | 4 | Medium | 2 | A |
| 28 | To create an output stream, we must declare the stream to be of class \_\_\_\_\_\_\_\_\_\_\_. | 1. ofstream 2. ifstream 3. iostream 4. None of these | 4 | Medium | 1 | A |
| 29 | Streams that will be performing both input and output operations must be declared as class \_\_\_\_\_\_\_\_\_. | 1. iostream 2. fstream 3. stdstream 4. Stdiostream | 4 | Medium | 1 | B |
| 30 | To perform IO manipulation which file has to be included into program | 1. fstream 2. iostream 3. iomanip | 4 | Hard | 1 | C |
| 31 | Which of the following is not used to seek a file pointer? | 1. ios::cur 2. ios::set 3. ios::end 4. ios::beg | 4 | Medium | 2 | B |
| 32 | The statement f1.write((char\*)&obj1, sizeof(obj1)); | 1. writes the member function of obj1 to f1 2. Writes the data in obj1 to f1. 3. Writes the member function and the data of obj1 to f1 4. Writes the address of obj1 to f1. | 4 | Easy | 1 | B |
| 33 | Which stream class is to only read from files? | 1. ofstream 2. ifstream 3. fstream 4. iostream | 4 | Easy | 1 | B |
| 34 | Which function is used in C++ to get the current position of file pointer in a file? | 1. tellp() 2. get\_pos() 3. get\_p() 4. tell\_pos() | 4 | Easy | 1 | A |
| 35 | Which among following is used to open a file in binary mode? | 1. ios:app 2. ios::out 3. ios::in 4. ios::binary | 4 | Medium | 2 | D |
| 36 | Which of the following is used to move the file pointer to start of a file? | 1. ios::beg 2. ios::start 3. ios::cur 4. ios::first | 4 | Medium | 2 | A |
| 37 | Which among following is correct syntax of closing a file in c++ ? | 1. myfile$close(); 2. myfile@close(); 3. myfile:close(); 4. myfile.close(); | 4 | Medium | 2 | D |
| 38 | What is use of eof() ? | 1. Returns true if a file open for reading has reached the next character. 2. Returns true if a file open for reading has reached the next word. 3. Returns true if a file open for reading has reached the end. 4. Returns true if a file open for reading has reached the middle. | 4 | Medium | 2 | C |
| 39 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_are used to format the data display in CPP? | 1. Iterators 2. Punctuators 3. Manipulators 4. Allocators | 4 | Medium | 3 | C |
| 40 | offset counted from the current position using? | 1. ios::curr 2. ios::cr 3. ios::cur 4. ios::current | 4 | Medium | 2 | C |
| 41 | Which of the followings is/are not a manipulator/s ?  1. flush 2. base 3. ends  4. oct 5. Bin 6. skipws | 1. Only 1, 6 2. Only 1,4,6 3. Only 1,3,6 4. Only 2,5 | 4 | Easy | 2 | D |
| 42 | Which of the following manipulator is used for the representing octal equivalent of a given decimal number? | 1. a. oct 2. setbase(8) 3. tobase(8) 4. both a and b | 4 | Medium | 3 | D |
| 43 | Which is among following is used to Open a file for output and move the read/write control to the end of the file? | 1. ios::ate 2. ios::at 3. ios::ann 4. ios::end | 4 | Medium | 3 | A |
| 44 | Which is correct syntax for, position n bytes back from end of fileObject ? | 1. fileObject.seekg(ios::end, n); 2. fileObject.seekg(n, ios:end ); 3. fileObject.seekg(n, ios::end ); 4. fileObject.seekg(ios:end, n); | 4 | Medium | 3 | C |
| 45 | How to find the position at end of fileObject? | 1. fileObject.seekg(0, ios::end ); 2. fileObject.seekg(0, ios::end ); 3. fileObject.seekg(0, ios::end ); 4. fileObject.seekg(0, ios::end ); | 4 | Hard | 4 | A |
| 46 | How to get position n bytes forward in fileObject? | 1. fileObject.seekg( ios::cur, n ); 2. fileObject.seekg( n, ios:cur ); 3. fileObject.seekg( n, ios::cur ); 4. fileObject.seekg( ios:cur, n ); | 4 | Hard | 3 | C |
| 47 | How to get position to the nth byte of fileObject? | 1. fileObject.seekg( 'filename',n ); 2. fileObject.seekg( n, 'filename' ); 3. fileObject.seekg( n ); 4. fileObject.seekg( n, ios::app ); | 4 | Easy | 3 | C |
| 48 | eof() is used to get | 1. easy code review 2. end of file 3. debug report 4. file close | 4 | Easy |  | B |
| 49 | Predict the output:  int x = 786;  cout <<setfill(‘\*’)<<setw(6)<<x; | 1. 786\*\*\* 2. \*\*786 3. \*\*\*786 4. \*\*\*\*\*\* | 4 | Medium | 1 | C |
| 50 | Unformatted IO functions are handled by\_\_\_\_ | 1. iostream 2. fstream 3. membuff 4. none of these | 4 | Medium | 3 | A |
| 51 | In the iostream library the ios class is at the root of the class hierarchy. | 1. True 2. False | 4 | Easy | 1 | A |
| 52 | Only for those manipulators which need an argument, we need to include the file ‘iomanip.h’ | 1. True 2. False | 4 | Easy | 1 | A |
| 53 | We can unset manipulators using unsetf() function. | 1. True 2. False | 4 | Easy | 1 | A |
| 54 | cout, cin & cprn are predefined stream objects. | 1. True 2. False | 4 | Easy | 1 | A |
| 55 | strstreams serve the same purpose as sprint() & sscanf() | 1. True 2. False | 4 | Medium | 1 | B |
| 56 | While writing binary data to a file we must specify delimiters to differentiate the data. | 1. True 2. False | 4 | Hard | 2 | A |
| 57 | What is command line arguments? | 1. Arguments passed to main() function 2. Arguments passed to any function 3. Arguments passed to class functions 4. Arguments passed to structure functions | 4 | Easy | 1 | A |
| 58 | What is the signature of main function using command line arguments? | 1. int main(int argc, char const \*argv[]) 2. int main(int argc, char const \*\*argv) 3. int main(int argc, char \*\*argv) 4. all of the above | 4 | Medium | 1 | D |
| 59 | Which character is used to separate different argument? | 1. # c. space 2. $ d. | | 4 | Easy | 1 | A |
| 60 | If we want to print output onto console then which port can be used? | 1. PRN c. LPT2 2. CON d. NUL | 4 | Easy | 1 | A |
| 61 | How to store the large objects in C++ if it extends its allocated memory? | 1. queue c. Memory heap 2. Stack d. stack & queue | 4 | Easy | 1 | A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Faculty: Mr. Hule Kuldeep A.**  **Name of Subject: Object Oriented Programming (210243)**  **Year and Branch: SE(COMP)** | | | | | | |
| **Q. No.** | **Description Question** | **Choices** | **Unit No** | **Difficulty levels (Easy/Medium/Hard)** | **BT Levels** | **Correct Answer** |
| **Unit- V Exception Handling & Templates** | | | | | | |
| 1 | Which of the following most preferred way of throwing and handling exception? | 1. Throw by value and catch by reference 2. Throw by reference and catch by value 3. Throw by value and catch by value 4. None of these | 5 | Easy | 2 | D |
| 2 | Which of the following is the most general exception handler that catches exception of any type? | 1. catch(std::exception) 2. catch(std:any\_exception) 3. catch(...) 4. catch() | 5 | Medium | 2 | C |
| 3 | The process of handling the actual exception occurs \_\_\_\_\_\_\_\_\_ | 1. inside the program 2. outside the program 3. both a &b 4. none of these | 5 | Easy | 1 | B |
| 4 | For handling the exception in C++ \_\_\_\_\_\_\_ are used | 1. catch handlers 2. exception handlers 3. Pointers 4. none of these | 5 | Medium | 2 | B |
| 5 | Choose the correct statement | 1. A function can throw any type of exception 2. a function can throw an exception of certain type only 3. A exception can't throw any type of exception 4. none of these | 5 | Easy | 2 | B |
| 6 | What function will be called when we have uncaught exception? | 1. Catch 2. throw 3. terminate 4. none of these | 5 | Medium | 1 | C |
| 7 | What will happen when a program throws any other of exception other than specified? | 1. still execute 2. Terminate 3. raise an error 4. none of these. | 5 | Medium | 1 | C |
| 8 | Which keyword can be used as a template | 1. exception 2. typename 3. both a & b 4. function | 5 | Easy | 1 | B |
| 9 | Which parameter is legal for non-type template? | 1. pointer to member 2. object 3. class 4. none of these | 5 | Easy | 1 | A |
| 10 | When template is defined with parameter that would be replaced by specified \_\_\_\_\_\_\_at the time of actual use of class or function. | 1. Keyword 2. Operator 3. Datatype 4. None of the above | 5 | Medium | 1 | C |
| 11 | Templates sometimes called as \_\_\_\_\_\_\_\_\_\_\_ | 1. Parameterized classes 2. Parameterized function 3. Both a and b 4. None of the above | 5 | Medium | 1 | A |
| 12 | Exceptions are of type | 1. Synchronous 2. Asynchronous 3. Both a and b 4. None of the above | 5 | Medium | 2 | C |
| 13 | “out-of-range”, “overflow” are the type of exceptions | 1. Asynchronous 2. Synchronous 3. Default 4. None of the above | 5 | Easy | 3 | A |
| 14 | The most type of error--------. | 1. Logical error 2. Syntactic error 3. Both a and b 4. Class | 5 | Easy | 1 | B |
| 15 | Can we throw exception more than one time | 1. Yes 2. No 3. Can’t say | 5 | Easy | 1 | A |
| 16 | Which statement we have to use rethrowing exception | 1. throw(exception) 2. throw 3. Both a and b 4. None of the above | 5 | Easy | 1 | C |
| 17 | Class template is applicable for \_\_\_. | 1. For function only 2. For that class only 3. Both a and b 4. None of the above | 5 | Medium | 2 | C |
| 18 | Which type of program is recommended to include in try block | 1. Static memory allocation 2. Dynamic memory allocation 3. Const reference 4. Pointer | 5 | Easy | 1 | B |
| 19 | What is meaning of template parameter? | 1. It is used to pass a type as argument 2. Used to evaluate a type 3. It can of no return type 4. None of the mentioned | 5 | Hard | 2 | A |
| 20 | \_\_\_\_\_\_Keyword is used in template. | 1. class 2. typename 3. Both a and b 4. using | 5 | Easy | 1 | C |
| 21 | Generic programming is approach of\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ which are applicable for all types | 1. Generalized algorithm 2. Pseudo algorithm 3. Both a and b 4. None of the above | 5 | Hard | 3 | D |
| 22 | Template function can be overloaded | 1. True 2. False | 5 | Easy | 6 | A |
| 23 | Pick up the correct statement  i)template allow us to define generic classes and functions  ii)template support generic programming  iii)function template overloading is possible | 1. i only 2. i and ii only 3. ii and iii only 4. i, ii and iii | 5 | Hard | 3 | D |
| 24 | Syntax for creating a function template is | 1. Template<typename t> returntype function name 2. Template<class T> returntype function name 3. Both a and b 4. None of the above mentioned | 5 | Medium | 2 | C |
| 25 | Actual code for a template function is generated when | 1. the function declaration appears in the source code. 2. the function definition appears in the source code. 3. a call to the function appears in the source code. 4. the function is executed at runtime. | 5 | Medium | 2 | C |
| 26 |  | 1. Start   Inside try block  Caught an exception -- value is:100  End   1. Start   End   1. Start   Inside try block  End   1. None of the above mentioned | 5 | Easy | 2 | A |
| 27 | The compiler generates only one version of function template for each data type irrespective of the number of calls that are made for that type. | 1. True 2. False | 5 | Medium | 2 | A |
| 28 | Using templates saves memory. | 1. True 2. False | 5 | Medium | 1 | A |
| 29 | We cannot inherit a new class from class template. | 1. True 2. False | 5 | Hard | 2 | B |
| 30 | A class template member function can be defined outside the class template. | 1. True 2. False | 5 | Easy | 1 | A |
| 31 | Formal parameter names must be unique in the parameter list of a function template. | 1. True 2. False | 5 | Medium | 2 | B |
| 32 | We can distribute function templates & class templates in object libraries. | 1. True 2. False | 5 | Easy | 1 | A |
| 33 | Template arguments can take default values. | 1. True 2. False | 5 | Easy | 2 | A |
| 34 | The exception handling mechanism is supposed to handle compile time errors. | 1. True 2. False | 5 | Easy | 1 | B |
| 35 | It is always necessary to declare the exception class within the class in which an exception is going to be thrown. | 1. True 2. False | 5 | Medium | 2 | B |
| 36 | For one try block there can be multiple catch blocks. | 1. True 2. False | 5 | Hard | 2 | A |
| 37 | When an exception is thrown an exception class’s constructor gets called. | 1. True 2. False | 5 | Easy | 2 | A |
| 38 | In a program if there is a possibility of an exception then it is necessary write all the statements in the try block. | 1. True 2. False | 5 | Medium | 1 | B |
| 39 | Try blocks cannot be nested. | 1. True 2. False | 5 | Easy | 5 | A |
| 40 | Proper destruction of an object is guaranteed by exception handling mechanism. | 1. True 2. False |  | Easy | 2 | A |
| 41 |  | 1. 0.23 10   X Template add power   1. 10 10   X template add power   1. Compilation error 2. 10 0.23   X Templates add power. | 5 | Medium | 5 | D |
| 42 |  | 1. 6 10 2. 5 5 3. 10 10 4. Compilation error | 5 | Medium | 5 | A |
| 43 |  | 1. Exception happened 2. My exception happened. 3. Run Time error 4. Compilation error | 5 | Medium | 5 | B |
| 44 | Pick up the correct statement from following  1.Exception handling is not supported c++  2.Template support generic programming in c++  3.overloading of function template is possible in c++  4.generic catch template can handle all types of exceptions | 1. 2 and 3 only 2. 3 and 4 only 3. 1, 2 and 3 only 4. 2, 3 and 4 only | 5 | Medium | 2 | D |
| 45 |  | 1. 2.1 2. 3.1 3. 4.41 9.61 16.81 4. 2.1 3.1 4.1 | 5 | Medium | 5 | C |
| 46 | Why we use :: template-template parameter? | 1. binding 2. rebinding 3. both binding & rebinding 4. reusing | 5 | Medium | 2 | C |
| 47 | Which parameter is legal for non-type template? | 1. pointer to member 2. object 3. class 4. baseclass | 5 | Easy | 3 | A |
| 48 | Templates are abstract recipe for producing a concrete code, and it is used for | 1. Producing functions 2. Producing classes 3. Nothing 4. Both A and B | 5 | Easy | 2 | D |
| 49 | Pick out the correct statement about string template? | 1. It is used to replace a string. 2. It is used to replace a string with another string at runtime. 3. It is used to delete a string. 4. none of the mentioned | 5 | Easy |  | A |
| 50 |  | 1. Null pointer on second type-cast 2. Null pointer on first type-cast 3. Null pointer on third type-cast 4. Exception: NULL pointer exception | 5 | Medium | 5 | A |
| 51 |  | 1. 100 2. 3.1416 3. 100   3.1416   1. 4.14 | 5 | Easy | 5 | C |
| 52 |  | 1. unexpected handler called 2. caught bad\_exception 3. caught other exception 4. both unexpected handler called and caught bad\_exception | 5 | Easy | 5 | A, B |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name of Faculty: Mr. Hule Kuldeep A.**  **Name of Subject: Object Oriented Programming (210243)**  **Year and Branch: SE(COMP)** | | | | | | |
| **Q. No** | **Description Question** | **Choices** | **Unit No** | **Difficulty levels (Easy/Medium/Hard)** | **BT Levels** | **Correct Answer** |
| **Unit- VI Standard Template Library (STL)** | | | | | | |
| 1 | What is the Standard Template Library? | 1. Set of C++ template classes to provide common programming data structures and functions 2. Set of C++ classes 3. Set of Template functions used for easy data structures implementation 4. Set of Template data structures only | 6 | Easy | 2 | A |
| 2 | Pick the correct statement. | 1. STL is a generalized library 2. Components of STL are parameterized 3. STL uses the concept of templates classes and functions to achieve generalized implementation 4. All of the mentioned | 6 | Medium | 2 | D |
| 3 | How many components STL has? | 1. 1 c. 2 2. 3 d. 4 | 6 | Easy | 2 | B |
| 4 | 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 | 6 | Medium | 2 | A |
| 5 | In how many categories, containers are divided? | 1. 1 c. 2 2. 3 d. 4 | 6 | Medium | 2 | B |
| 6 | 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 | 6 | Medium | 2 | A |
| 7 | How many Sequence Containers are provided by C++? | 1. 2 c. 3 2. 4 d. 5 | 6 | Easy | 2 | C |
| 8 | What are the Associative 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 | 6 | Medium | 2 | C |
| 9 | How many Associative Containers are provided by C++? | 1. 2 c. 4 2. 3 d. 5 | 6 | Easy | 1 | C |
| 10 | How many Container Adaptors are provided by C++? | 1. 2 c. 4 2. 3 d. 5 | 6 | Medium | 2 | B |
| 11 | What are Unordered Associative 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 | 6 | Easy | 1 | C |
| 12 | What are Container Adaptors? | 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 provide a different interface for sequential containers | 6 | Medium | 2 | D |
| 13 | What are Iterators? | 1. Iterators are used to iterate over C-like arrays 2. Iterators are used to iterate over pointers 3. Iterators are used to point memory addresses of STL containers 4. Iterators are used to iterate over functions | 6 | Medium | 2 | C |
| 14 | How many types of Iterators are provided by C++? | 1. 2 c. 3 2. 4 d. 5 | 6 | Medium | 2 | D |
| 15 | Which header file is used for Iterators? | 1. <iterator> c. <algorithm> 2. <iter> d. <loopIter> | 6 | Medium | 2 | A |
| 16 | C++: A collection of generic class and function is called as | 1. Standard Template Library(STL) 2. Header file 3. Function template. 4. d. None of the above. | 6 | Easy | 1 | A |
| 17 | Meaning of STL | 1. Standard Tree Library 2. Standard Term Library 3. Standard Template Library 4. None of above | 6 | Medium | 2 | C |
| 18 | Meaning of deed copy is | 1. A deep copy creates a copy of the dynamically allocated objects too 2. A deep copy just copies the values of the data as they are 3. A deep copy creates a copy of the statically allocated objects too 4. Both A and C | 6 | Medium | 2 | A |
| 19 | Which of the following is true about templates? | 1. Template is a feature of C++ that allows us to write one code for different data types. 2. We can write one function that can be used for all data types including user defined types. Like sort(), max(), min(), ..etc. 3. We can write one class or struct that can be used for all data types including user defined types. Like Linked List, Stack, Queue ..etc. 4. Template is an example of compile time polymorphism. | 6 | Medium | 2 | C |
| 20 |  | 1. x=1 count=0   x=1 count=1  x=1.1 count=1   1. x=1 count=0   x=1 count=0  x=1.1 count=0   1. x=1 count=0   x=1 count=1  x=1.1 count=2   1. Compiler Error | 6 | Hard | 4 | A |
| 21 |  | 1. 7   7.0  7.0   1. Compiler Error in all cout statements as data type is specified 2. Compiler Error in last cout statement as call to max ambiguous 3. None of these | 6 | Hard | 5 | D |
| 22 |  | 1. 0   0   1. 1   1   1. 2   1   1. 1   0 | 6 | Hard | 4 | C |
| 23 |  | 1. 16   24   1. 8   16   1. 20   28   1. Compiler Error: template parameters are not default | 6 | Hard | 4 | WRONG OPTIONS CORRECT ANSWER  2  8 |
| 24 |  | 1. Compiler Error: template argument must be a data type 2. 10   1   1. 1000   256   1. 1   1 | 6 | Hard | 5 | B |
| 25 |  | 1. 10 2. 20 3. Compiler Error 4. 30 | 6 | Hard | 4 | C |
| 26 |  | 1. 20 2. Called 20 3. Compiler Error 4. Called 10 | 6 | Hard | 2 | B |
| 27 |  | 1. Compiler Error 2. 1024 3. 2 4. 1 | 6 | Hard | 4 | B |
| 28 | Which container is used to store elements as key-value pair? | 1. map 2. multimap 3. unordered map 4. all of the mentioned | 6 | Medium | 2 | D |
| 29 | Which container can have the same keys? | 1. map 2. multimap 3. unordered map 4. set | 6 | Medium | 2 | B |
| 30 | Which container is best to keep the collection of distinct elements? | 1. multimap 2. heap 3. set 4. queue | 6 | Medium | 2 | C |
| 31 | Which container is used to keep priority-based elements? | 1. queue 2. stack 3. set 4. priority queue | 6 | Medium | 2 | D |
| 32 | Sets are implemented using \_\_\_\_\_\_\_\_\_\_ | 1. binary search tree 2. red black tree 3. avl tree 4. heap | 6 | Easy | 1 | B |
| 33 | Unordered map is implemented using \_\_\_\_ | 1. binary search tree 2. red black tree 3. heap 4. hash table | 6 | Easy | 1 | D |
| 34 | Map is implemented using \_\_\_\_\_ | 1. binary search tree 2. red black tree 3. heap 4. hash table | 6 | Easy | 1 | B |
| 35 | Which of the following is correct about the map and unordered map? | 1. Ordering of keys in maps whereas no such order in the unordered map 2. Maps are implemented red-black trees whereas unordered map are implemented using hash tables 3. Average search time in the unordered map is O(1) whereas it is O(logn) in case of maps 4. All of the mentioned | 6 | Medium | 2 | D |
| 36 | Which of the following queue container can expand or shrink from both directions? | 1. deque 2. queue 3. priority queue 4. stack | 6 | Medium | 2 | A |
| 37 | Which of the following is correct about map and multimap? | 1. Map can have same keys whereas multimap cannot 2. Implementation of maps and multimap are different 3. Multimap can have same keys whereas the map cannot 4. Average search time of map is greater than multimap | 6 | Medium | 2 | C |
| 38 | What is any in C++? | 1. STL container used to store a single value of any type 2. Exception class in C++ 3. Fundamental type provided by C++ 4. Template data type | 6 | Easy | 2 | A |
| 39 | In how many different ways any-container can be constructed? | 1. 1 c. 3 2. 2 d. 4 | 6 | Medium | 2 | C |
| 40 | What is the correct syntax of constructing any using copy initialization? | 1. any variable\_name = value; 2. any variable\_name(value); 3. any <type>variable\_name = value; 4. none of these | 6 | Medium | 2 | A |
| 41 | Which of the following syntax is used to convert any variable to its original type? | 1. any\_cast<variable\_name>(); 2. any\_cast(variable\_name); 3. <original\_type>(variable\_name); 4. any\_cast<original\_type>(variable\_name); | 6 | Easy | 3 | C |
| 42 | Which header file is required to use any container? | 1. <any> 2. <stl> 3. <container-any> 4. <containers> | 6 | Easy | 2 | A |
| 43 | What are Iterators? | 1. STL component used to point a memory address of a container 2. STL component used for vectors 3. STL component used to call functions efficiently 4. STL component used to define template classes | 6 | Easy | 1 | A |
| 44 | Which function is used increment the iterator by a particular value? | 1. next() c. prev() 2. advance() d. move() | 6 | Easy | 2 | B |
| 45 | Pick the correct statement. | 1. Input iterator moves sequentially forward 2. Input iterator moves sequentially backward 3. Input iterator moves in both direction 4. Input iterator moves sequentially downwards | 6 | Medium | 2 | A |
| 46 |  | 1. 2 2. 3 3. 4 4. 5 | 6 | Hard | 4 | B |
| 47 |  | 1. 3 2. 4 3. Error 4. Segmentation fault | 6 | Easy | 4 | C |
| 48 |  | 1. 5 2. Compile-time error 3. Run-time error 4. Nothing is printed | 6 | Medium | 3 | B |
| 49 |  | 1. Run-time error 2. Compile-time error 3. Hello World 4. Nothing is printed | 6 | Medium | 2 | B |
| 50 |  | 1. Hello World 2. Compile-time error 3. Run-time error 4. Nothing is printed | 6 | Medium | 2 | C |
| 51 | When defining an iterator from the STL, the compiler automatically creates the right kind, depending upon the container it is to be used with. | 1. True 2. False | 6 | Easy | 2 | A |
| 52 | STL algorithms are implemented as function templates | 1. True 2. False | 6 | Easy | 2 | A |
| 53 | STL generally avoids generic programming in favor of using inheritance and polymorphism to achieve better execution-time performance. | 1. True 2. False | 6 | Easy | 2 | B |
| 54 | In C and in raw C++, we access elements of an array with pointers. In C++ STL, we access elements of containers via iterator objects that have the "feel" of pointers, but behave more intelligently. | 1. True 2. False | 6 | Medium | 2 | A |
| 55 | Pointers into arrays may be used in place of iterators in most STL algorithms, including those that require random-access iterators. | 1. True 2. False | 6 | Medium | 2 | A |
| 56 | All associative containers except multiset support bidirectional iterators | 1. True 2. False | 6 | Easy | 2 | B |
| 57 | STL algorithms are implemented as member functions of the STL containers. | 1. True 2. False | 6 | Easy | 2 | D |
| 58 | Each of the common operations of a stack, queue, and priority\_queue are implemented as inline functions that call the appropriate function of the underlying container. | 1. True 2. False | 6 | Hard | 2 | A |
| 59 | The STL stack container is an adapter for the vectors, lists, deques STL containers. | 1. True 2. False | 6 | Easy | 2 | A |
| 60 | Linked list are not superior to STL vectors. | 1. True 2. False | 6 | Easy | 2 | A |

🙦🙦🙦 ***All the Best for the Exam*** 🙤🙤🙤