Fall 2022
CS304 - Object Oriented Programming
Final Term Sample Paper

1. What is the keyword used in templates in C++?

   A. class
   B. typename
   C. both class & typename (**Correct**)
   D. function

2. What is the purpose of using the "::" template-template parameter in C++?

   A. Binding
   B. Rebinding
   C. Both binding & rebinding (**Correct**)
   D. Reusing

3. What does not require instantiation in C++?

   A. Functions
   B. Non-virtual member function
   C. Member class
   D. All of the mentioned (**Correct**)

4. What is the keyword used to define user-defined data types in C++?

   A. def
   B. union
   C. typedef (**Correct**)
   D. type

5. What is the scope of data types defined with typedef in C++?

   A. Inside that block only
   B. Whole program (**Correct**)
   C. Outside the program
   D. Main function

6. What is a template in C++?

   A. A formula for creating a generic class (**Correct**)
   B. A tool for manipulating a class
   C. A way of creating attributes
   D. None of the given

7. What is used for generic programming in C++?

   A. Modules
   B. Templates (**Correct**)

C. Virtual functions
D. Abstract Classes

8. What can be passed by non-type template parameters at compile time in C++?

A. int
B. float
C. Constant expression (**Correct**)
D. char

9. Function template in C++ is way of creating a function_____.

A. without a class
B. with an exact type
C. without blank spaces
D. without specifying the exact type (**Correct**)

10. Which of the following is not an advantage of using Templates in C++?

A. High Reliability **(Correct)**
B. Reusability
C. Writability
D. Generic Programming

11. Which one of the following is not a type of inheritance in C++?
A. public
B. virtual (**Correct**)
C. private
D. protected

12. In C++ inheritance, is it possible for a child class to initialize its indirect base classes through its constructor initialization list?
A. Yes, it is possible
B. No, it is not possible (**Correct**)
C. It depends on the implementation.
D. Only if the indirect base classes are direct base classes of the child class.

13. How many classes are required to implement the multiple inheritance in a program?
A. Only two
B. At least two
C. At least three (**Correct**)
D. Only four

14. _____ is used to solve multiple inheritance issues.
A. Virtual inheritance (**Correct**)
B. Friend functions
C. Generic programming.
D. Templates

15. What is multiple inheritance in Object-Oriented (OO) programming?

A. a derived class inherits from multiple base classes. (**Correct**)
B. a base class inherits from multiple derived classes.
C. a class inherits from a single base class.
D. a class cannot inherit from any base class.

16. A/an _____ is a template definition of methods and variables of a class and its object cannot be instantiated.
A. virtual class
B. static class
C. concrete class
D. abstract class(**Correct**)

17. Specialization (Restriction) can be implemented using both _____ and _____ inheritance.
A. Public, Protected
B. Private Protected (**Correct**)
C. Virtual, Protected
D. Public, Virtual
E.

18. Correct format to declare a template function is _____.
A. template< class T >
void funName(T x); (**Correct**)

B. template< classfunction T >
void funName(T x);

C. template< Function T >
void funName(T x);

D. template< class T function F>
void funName(T x);

19. The target function for a call is selected at run-time is known as_____.
A. Dynamic binding (**Correct**)
B. Static binding
C. Function binding
D. Object binding

20. A mechanism that enables us to write a single **function or class** that works for all data types is known as _____.

A. Generic programming (**Correct**)
B. Imperative programming
C. Functional programming
D. Logic programming

21. What is the purpose of using a cursor pointer declared outside an aggregate object?

A. Traverse elements (Correct)
B. Store fixed value
C. Store multiple values

D. Store address

22. Which of the following is not a method to traverse the element in a cursor?
    A. T* first()
    B. T* beyond()
    C. T* next(T*)
    D. T* prev(T*) (**Correct**)

23. Cursor works fine when we have _____ memory.
    A. Contagious (**Correct**)
    B. Non-contagious
    C. Static
    D. Dynamic

24. Which of the following is not a key component of STL (Standard Template Library)?
    A. Containers
    B. Iterators
    C. Algorithms
    D. Map (**Correct**)

25. Which of the following is not a Sequence Container?
    A. Iterator
    B. Deque
    C. Map(**Correct**)
    D. List

26. Which of the following is Not a type of Error handling technique?
    A. Graceful Termination
    B. Return an illegal value
    C. Return a value (**Correct**)
    D. Return code form a function

27. Which of the following follows the multi-pass algorithm?
    A. Input Iterator
    B. Output Iterator
    C. Random Access Iterator
    D. Bi-directional Iterator(**Correct**)

28. A copy of an element is made using _____ , when an element is inserted into a container.
    A. Copy constructor
    B. Assignment operator
    C. Copy constructor and Assignment operator (**Correct**)
    D. Destructor

29. _____ provides faster retrieval of data based on keys.
    A. Associative container (**Correct**)
    B. Sequence container
    C. Input Iterator
    D. Random Access Iterator

30. _____ can traverse the elements of a container without exposing the internal representation of the container.
    A. Cursor

B. Iterator (**Correct**)
C. Algorithm
D. Sequence Container

31. In order to overload **()** operator, the function must be a _____ function.

A. member **(Correct)**
B. static
C. Non-static
D. parameterized

32. Which of the following statement is not an using a unary operator?

A. --x
B. -(x++)
C. !(*ptr ++)
D. x+ptr (**Correct**)

33. The output of the follwing code snippet is _____.

```
int x = 1, y = 2;
cout << y++;
cout << y;
```

A. 23 (**Correct**)
B. 12
C. 21
D. 32

34. Which of the following is not the correct way to implement a post-increment operator?

A. y++ = x; (**Correct**)
B. y = x++;
C. x = y++;
D. y = y++;

35. Which of the following depicts the correct output of the following lines of code?

```
int x = 2, y = 2;
++++y;
cout << y;
++y = x;
cout << y;
```

A. 42 (**Correct**)
B. 22
C. 32
D. 23

36. Which of the following depicts the correct syntax of type conversion while operator overloading?

A. TYPE1 operator :: TYPE2();
B. TYPE1 :: operator TYPE2(); (**Correct**)
C. TYPE1 operator TYPE2 :: ();
D. TYPE1 :: TYPE2 operator();

37. In perspective of Inhertance, the casting from derived class to the base class is called _____.

A. Up casting (**Correct**)
B. Down casting
C. Dynamic casting
D. Polymorphic casting

38. If a constructor has some parameters having default values, the constructor is called _____ constructor.

A. Default (**Correct**)
B. Parameterized
C. Static
D. Non-static

39. The type that is used to declare a reference or pointer is called its _____ type.

A. Static (**Correct**)
B. Copy
C. Shallow copy
D. Deep copy

40. Conversions that compiler can perform automatically using built- in casting operations without casting request from programmer is known as _____ casting.

A. Implicit casting conversion (**Correct**)
B. Explicit casting conversion
C. Static casting conversion
D. Reinterpret casting conversion

==========

3 Mark Short Question

Question No. 41:

Write C++ code to define a template function that takes two parameters of the same type and returns the maximum of the two.

Note: The parameters and return types should be of generic type.

Solution:

```
template <typename T>
T maximum(T a, T b) {
  return (a > b) ? a : b;
}
```

or

```
template <typename T>
T maximum(T a, T b) {
  if (a > b) {
    return a;
  }
else {
    return b;
  }
```

```
}
```

Question No. 42:
What will be the output produced by this code?

```
#include <iostream>
using namespace std;

template <typename T>
class MyTemplateClass {
   private:
      T value;
   public:
      MyTemplateClass(T val) {
         value = val;
      }
      void displayValue() {
         cout << "Value: " << value << endl;
      }
};

int main()
{
   MyTemplateClass<int> intObj(10);
   MyTemplateClass<double> doubleObj(3.14);
   MyTemplateClass<char> charObj('A');

   intObj.displayValue();
   doubleObj.displayValue();
   charObj.displayValue();

   return 0;
}
```

Solution:
Output:
Value: 10
Value: 3.14
Value: A

Question No. 43:

Write down the problems we face while implementing multiple inheritance?

Solution:
If more than one base class have a function with same signature, then the child will have two copies of that function. Calling such function will result in ambiguity.

Question No. 44:
Write a template function named “isEqual()” which takes two generic input parameters and returns true if two values are equal or false if two values are not equal. The return type for function should be Boolean.

Solution:

```
template<typename T>
bool isEqual(T x, T y)
{
return (x==y);
}
```

Question No. 45:

What will be the output of the following C++ code?

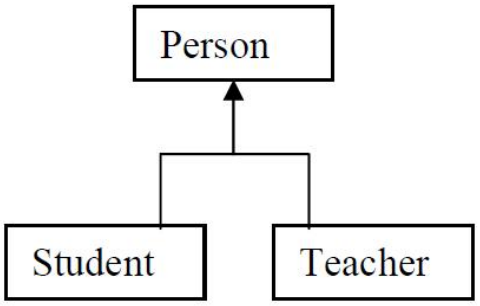
```
#include <iostream>
using namespace std;

int main() {
   double num_double = 3.56;
   cout << "num_double = " << num_double << endl;

   int num_int1 = (int)num_double;
   cout << "num_int1   = " << num_int1 << endl;

   int num_int2 = int(num_double);
   cout << "num_int2   = " << num_int2 << endl;

   return 0;
}
```

Solution:

```
num_double = 3.56
num_int1   = 3
num_int2   = 3
```

Question No. 46:

Write the code of classes shown in the following diagram.



Solution:

```
class Person{
};
```

```
class Teacher: public Person{
};


class Student: public Person{
};
```

==========

5 Marks Descriptive Question

Question No. 47:

Write C++ code for a **Template** class named **Rectangle** with private member variables **length** and **width**, and public member functions **setLength()** and **setWidth()** to set the values of **length** and **width**, and a member function **getArea** to calculate and return the **area** of the rectangle.

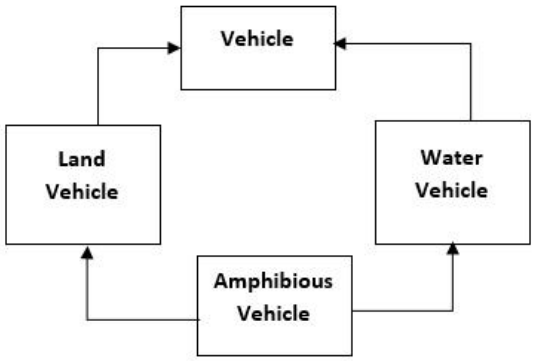
Note: length and width should be of generic type.

Solution:

```
template <typename T>
class Rectangle {
 private:
  T length;
  T width;

 public:
  void setLength(T l) {
    length = l;
  }

  void setWidth(T w) {
    width = w;
  }

  T getArea() {
    return length * width;
  }
};
```

Question No. 48:

Consider the given below class diagram and write the code to implement the following scenario.



Solution:

```
class Vehicle{
};
class LandVehicle : public Vehicle{
};

class WaterVehicle : public Vehicle{
};

class AmphibiousVehicle: public LandVehicle, public WaterVehicle{
}
};
```

Question No. 49:
Write a program which contains an integer vector to store the following value using the push_back() function.
12, 50, 32

Note: Include the required header file.

Solution:

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
        vector<int> v;
        v.push_back(12);
        v.push_back(50);
        v.push_back(32);

}
```

Question No. 50:
What will be the output of the following C++ Code?

```
#include<iostream>
using namespace std;
class Parent{
```

```
public:
Parent(){
cout<<"Parent Constructor...\n";
}
};
class Child : public Parent{
public:
Child(){
cout <<"Child Constructor...";}
};

int main(){
Child cobj;
return 0;
}
```

Solution:

```
Parent Constructor...
Child Constructor...
```