UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 2, Examination-Model Answers, July 2022

Module No: EE2201 **Module Name: Computer Programming II**

Part I

Instructions for candidates

- Write your index number on top of every page.
- Question paper contains 52 multiple choice questions.
- Each question carries 0.5 marks.
- Answer all questions. Each question has only one answer.
- Read the question and all answers before making the choice.
- **For each question, put an X mark on the letter: (a), (b), (c), or (d) which corresponds to the correct answer, by using a black or blue pen.**
- Time allowed is **1 hour and 30 minutes.**

1. What is a **class**?

   (a) User-defined data type
   (b) Object
   (c) Variable
   (d) Constant

2. A **class** contains

   (a) variables and functions
   (b) variables only
   (c) functions only
   (d) data structures and algorithms

Listing 1: Note: Table is a class

```
Table t1;
int lg;
t1.legs=4;
lg = t1.getLegs();
```

3. In Listing 1, the statement Table t1;

   (a) generates a compile time error
   (b) generates a run time error
   (c) creates an object t1
   (d) defines a function named t1

4. In Listing 1, legs must be a

   (a) member variable
   (b) member function
   (c) overloaded operator
   (d) template

5. In Listing 1, getLegs() must be a

   (a) member variable
   (b) member function
   (c) overloaded operator
   (d) template

Listing 2: Note: Car is a class

```
Car c1, *p;
p = &c1;
c1.engineCapacity =1500;
```

6. In Listing 2, the statement
   c1.engineCapacity =1500;
   can be written by using the pointer p as

(a) p−>engineCapacity = 1500;

(b) p::engineCapacity = 1500;

(c) p.engineCapacity = 1500;

(d) p:engineCapacity = 1500;

7. In Listing 2, the statement p = &c1;

(a) assigns the address of c1 to p

(b) means that the address of p is equal to the address of c1

(c) assigns the address of p to c1

(d) has a syntax error

Listing 3:

```
class cPoint
{ private:
   double x;
   double y;
  public:
   double getDist()
   {  return sqrt(x*x + y*y);
   }
   void setXY(double nx,
              double ny)
   {  x = nx; y = ny;
   }
};

int main()
{
    cPoint p;
    p.setXY(3,4);
    cout << p.getDist()<<endl;
    return 0;
}
```

8. In Listing 3, cPoint is a

(a) object

(b) variable

(c) method

(d) class

9. In Listing 3, p is a

(a) object

(b) variable

(c) method

(d) class

10. If executed, program in Listing 3 would display

(a) 25

(b) 7

(c) 49

(d) 5

11. In Listing 3, variable x can not be accessed in the main function. This statement is

(a) true

(b) false

(c) unclear

(d) not recognized in this context

12. In Listing 3, the key word **public** makes

(a) variables accessible outside the scope of the class.

(b) functions accessible outside the scope of the class.

(c) variables inaccessible within the scope of the class.

(d) functions inaccessible outside the scope of the class.

13. In Listing 3, the key word **private** makes

(a) member variables accessible outside the scope of the class.

(b) member functions accessible outside the scope of the class.

(c) member variables inaccessible outside the scope of the class.

(d) member functions inaccessible outside the scope of the class.

Listing 4:

```
class M
{
  public:
    M();
    M(double x11, double x12,
      double x21, double x22);
    M(M &m);
    ~M();
    void print();
    M & operator=(M &m);
    M & operator+(M &m);

  private:
    double x[4];
};
```

14. In Listing 4, the function M() is the

    (a) overridden default constructor
    (b) destructor
    (c) overloaded constructor
    (d) overloaded operator

15. In Listing 4, the function M(M&m) is the

    (a) overridden default constructor
    (b) destructor
    (c) overloaded constructor
    (d) overloaded assignment operator

16. In Listing 4, the function ˜M() is the

    (a) overridden default constructor
    (b) destructor
    (c) overloaded constructor
    (d) overloaded assignment operator

17. In Listing 4, the function
    M& **operator**=(M &m);
    is the

    (a) overridden default constructor
    (b) destructor
    (c) overloaded constructor
    (d) overloaded assignment operator

18. If m1 and m2 are objects of type M (defined in Listing 4), then which of the following statements has a compile time error?

    (a) m1;
    (b) m1+m2;
    (c) m1=m1+m2;
    (d) m1−m2;

19. If m1, m2 and m3 are objects of type M (defined in Listing 4), then which of the following statements has a compile time error?

    (a) m1+m2;
    (b) m1!=m2;
    (c) m1=m2=m3;
    (d) m1=m1+m2;

20. What is not an access specifier?

    (a) **private**
    (b) **public**
    (c) **protected**
    (d) lstinline*restricted*

21. Base classes in inheritance is also called

    (a) relation
    (b) top
    (c) child
    (d) parent

22. Derived classes in inheritance is also called

    (a) relation
    (b) top
    (c) child
    (d) parent

Listing 5:

```
class cShape
{ public:
    cShape();
    cShape(double x, double y);
    virtual ˜cShape();
    void Info();

  protected:
    double x;
    double y;

  private:
    int z;
};

class cCircle : public cShape
{
  public:
    cCircle();
    cCircle(double px ,
            double py );
    virtual ˜cCircle();
};
```

23. The variable z in Listing 5 is

    (a) **public** members
    (b) **private** members
    (c) **int** members
    (d) **friend** members

24. The variables x and y in Listing 5 act as

    (a) **public** members
    (b) **private** members

(c) **int** members

(d) **friend** members

25. Which statement is correct about the classes defined in Listing 5?

(a) cShape is the base class and cCircle is the derived

(b) cShape is the derived and cCircle is the base class

(c) cShape and cCircle both are base clases

(d) cShape and cCircle both are derived classes

26. Inheritance type of cCircle in Listing 5 is

(a) **private**

(b) **public**

(c) **protected**

(d) unknown

27. Member variables x and y are .... within the scope of cCircle (in Listing 5).

(a) restricted

(b) **protected**

(c) **public**

(d) **private**

28. Member variable .... is not inherited by cCircle (in Listing 5).

(a) w

(b) x

(c) y

(d) z

29. Which keyword is used to declare virtual functions?

(a) delegate

(b) virtual

(c) anonymous

(d) virtually

30. Virtual function is ...... class function which expected to be redefined in ...... class.

(a) base, derived

(b) derived, derived

(c) base, base

(d) derived, base

31. Virtual function should be a ...... member of the base class.

(a) public

(b) private

(c) protected

(d) public, private or protected

32. What is meant by multiple inheritance?

(a) Deriving a base class from derived class

(b) Deriving a derived class from base class

(c) Deriving a derived class from more than one base class

(d) None of the mentioned.

33. What inheritance allow in C++ program?

(a) Class re-usability

(b) Creating a hierarchy of classes

(c) Extendibility

(d) All of the above

34. What is a template?

(a) A template is a formula for creating a generic class

(b) A template is used to manipulate the class

(c) A template is used for creating the attributes

(d) Non of the above

35. When the inheritance is **public**, the private methods in base class are ..... in the derived class (in C++)

(a) inaccessible

(b) accessible

(c) protected

(d) public

36. What is the false statement about C and C++?

(a) C is a procedural while C++ is both procedural and object driven

(b) File extensions of C and C++ source files are *.c and *.cpp respectively

(c) Access modifiers are available only in C++.

(d) Pointers are available only in C.

37. What is the false statement about C and C++?

(a) Only C++ has direct support for exception handling.

(b) C does not support operator overloading.

(c) Both C and C++ supports dynamic memory allocation.

(d) C supports reference variables.

38. If a header file is defined in a C++ source file as
#define "myheader"
then it must be a

(a) user defined header file.

(b) native C++ header file

(c) native C header file

(d) header file from another language other than C++.

39. A C++ program statement can be commented by using

(a) // single line comment.

(b) \\ single line comment.

(c) */ .. */ multi-line comment.

(d) /* .. /* multi-line comment.

40. A **namespace** is a declarative region that provides a scope to the

(a) program

(b) exceptions

(c) classes

(d) identifiers

41. Without the declaration
using namespace std;
the text Hello, C++ can be displayed by

(a) cout<<"Hello, C++";

(b) std :: cout<<"Hello, C++";

(c) cout>>"Hello, C++";

(d) std:cout>>"Hello, C++";

42. In the statement
cin >>hello;

(a) hello is a variable.

(b) cin is an operator.

(c) >> should be corrected as <<.

(d) ; at the end is not necessary.

43. After execution of the following statements
double door = 175.5;
double &x= door;
x=25;

(a) x becomes 174.5.

(b) door becomes double the value of 175.5.

(c) x and doors swap their values.

(d) door becomes 25.

44. A source file has multiple functions with same name. These may be

(a) delegated functions.

(b) anonymous functions.

(c) overridden functions.

(d) overloaded functions.

45. The function to be executed among overloaded functions is selected based on the

(a) return type.

(b) input parameters.

(c) body content.

(d) position in the source file.

46. The operator introduced in C++ for dynamic memory allocation is

(a) malloc

(b) new

(c) realloc

(d) mget

47. The operator introduced in C++ for deleting the dynamically allocation memory is

(a) realloc

(b) delete

(c) free

(d) mget

48. What is the correct way to dynamically allocate a variable of type **int**?

    (a) **int** *x = **new int**;

    (b) **int** y = **new int** [10];

    (c) **int** *z = **int new**;

    (d) **int new** *z;

49. What is the correct way to dynamically allocate 10 variables of type **double**?

    (a) **double** x[10] = **double int**;

    (b) **double** *y = **new double** [10];

    (c) **double** *z = **double** [10] **new**;

    (d) **double new** z[10];

50. What is the correct way to deallocate, dynamically allocated array grades of type **int**?

    (a) **delete** [] grades;

    (b) **delete** grades [];

    (c) **delete** [10] grades;

    (d) free grades[10];

51. Who is the founder of the C++ language?

    (a) Anders Hejlsberg

    (b) James Gosling

    (c) Brendan Eich

    (d) Bjarne Stroustrup

52. A base class called Animal has a method called animalSound(). Derived classes of Animals are Pigs, Cats, Dogs, Birds, and they also have their own implementation of an animal sound by a method of same name (that is animalSound()). This examples shows ..... of object oriented programming.

    (a) encapsulation

    (b) polymorphism

    (c) security

    (d) under performance