

C++ Final Exam

Duration: 0.5 hr.

Name :

Group :

Choose only one answer.

1. void main()

```
{  
    int friend;  
    friend+=5;  
}
```

The above code is

- a) C valid only.
- b) C and C++ valid.
- c) C++ valid only.
- d) **C and C++ invalid**

2. int a;

void main()

```
{  
    int a;  
    a=50;  
    ::a=20;  
    cout << a;  
}
```

The above piece of code will print:

- a) The value of 20.
- b) Compiler will give an error.
- c) **The Value of 50.**
- d) None of above.

3. The following function prototype uses the default argument

```
void myfunc(int x=3,int y);
```

Is the above function valid?

- a) Yes, you can give a default parameter for x and not for y.
- b) No, both x and y must have a default values.
- c) No, you can give a default parameter for y and not for x.
- d) **No, there must be no arguments without defaults to the right of the default arguments.**

4. void main()

```
{  
    int *p;  
    p=new int(5);  
}
```

The above code

- a) Allocates an integer that is not initialized.
- b) Allocates an array of integers of 5 elements that is not initialized.
- c) **Allocates an integer that is initialized by 5.**
- d) Allocates an array of integers where all its elements is initialized by 5.

5. class myclass

```
{  
    int x;  
protected:  
    int y;  
};  
void main()  
{  
    myclass c1;  
    c1.x=100;  
    c1.y=500;
```

```
}
```

The above code will produce

- a) An error, x is private.
- b) An error, y is protected.
- c) No error
- d) **Both (a) and (b)**

6. The class constructor must

- a) Return void.
- b) **Have the same name as class.**
- c) Have no parameter.
- d) Both (a) and (b).

7. Which type of stand alone functions is allowed to access the private members of a class?.

- a) Template Function.
- b) Inline Function.
- c) **Friend Function.**
- d) Overloaded Function.

8. class base

```
{  
    int x;  
protected:  
    int y;  
public:  
    int c;  
};  
class derived:private base  
{  
    //some code  
};
```

Class derived can access:

- a) Private and protected members of base class.

- b) Only private members of base class.
- c) Only protected members of base class.
- d) **Protected and public members of the base class.**

9. class base

```
{  
public:  
    base(){cout << "Constructing base class\n";}  
};  
class derived:public base  
{  
public:  
    derived(){cout << "Constructing derived class\n";}  
};  
void main()  
{  
    derived d;  
  
  
  
  
}
```

The following code will print:

- a) Constructing base class
- b) **Constructing base class**
Constructing derived class
- c) Constructing derived class
- d) Constructing derived class
Constructing base class

10. class base

```
{  
public:  
    void say() {cout << "\n I'm base \n";}  
};  
class derived:public base
```

```
{
public:
    void say() {cout << "\n I'm derived \n";}
};
void main()
{
    base *pb;
    base mybase;
    derived myderived;

    pb=&mybase;
    pb->say();
    pb=&myderived;
    pb->say();
}
```

The previous code will print:

- a) I'm base
I'm derived
- b) I'm base
I'm base
- c) I'm derived
I'm derived
- d) Non of the above

11. Passing reference to object as a function parameter in order to

قصدة المنظر دة كدة

Calss ThreeD { }

Void myfunc (ThreeD & t);

- a) The destructor not to be called
- a) The constructor not to be called

- b) Make a call by value function call
 - c) Have no effect
12. Operator Overloading
- a) Changes the precedence of the operator
 - b) **Extends the functionality of the operator**
 - b) Changes the default function of the operator
 - c) Has no effect on the operator
13. The protected members for the class user is equivalent to
- a) Public Member
 - c) **Private Member**
 - d) Global Variable
 - b) Local Variable
14. The constructor is called with
- a) Declaring a class
 - b) **Defining an object of the class**
 - c) Calling the constructor function
 - d) Removing the object from the program memory
15. The class attribute can be characterized by:
- a) **Every object has it's own value of the class attribute**
 - b) All objects of the class have the same value
 - c) Make the variable scope within the class
 - d) Illegal definition
16. The virtual base class
- a) Can be accessed through a pointer to base class
 - b) Can not create an object of it.
 - c) **Is used to avoid multiple instance in multiple inheritance**
 - d) Which contains pure virtual functions
17. Structure Objects are معرفش أية دة

- a) Objects holding information of system
- b) Objects deals with file data structures
- c) Objects support communication with users
- d) All of the above

18. Aggregation is

- a) A special type of inheritance
- b) A special type of association
- c) Is similar to association
- d) Is similar to inheritance

20. A one way association with multiplicity many is represented by:

زى كدة

Clas Rectangle {

Public:

Rectangle(Point * p1 , Point * p2);

}

يعنى البوينترز بتبقى فى جانب واحد أو فى كلاس واحد فى حالة الأسوسياشن

- a) One pointer in the one side
- b) Set of pointers in the one side
- c) One pointer in the either sides.
- d) Set of pointer in the either sides.