Q1. #include<iostream>

using namespace std;

class Sample

{

static int count;

public:

void show()

{

cout<<"count="<<count++;

}

};

int Sample :: count=0;

int main()

{

Sample s;

cout<<sizeof(s);

}

What would be the output of the code??

**A. 1**

B. 4

C. Compile time error.

D. No output

Q2. Which is the correct syntax for returning an object by value?

a. void functionName ( ){ }

b. void functionName ( ){ }

c. class object functionName( ) { }

**d. ClassName functionName ( ){ }**

Q3. #include<iostream>

#include<cstring>

using namespace std;

class Sample

{

private:

int num;

char name[20];

public:

Sample(int x,char n[])

{

num=x;

strcpy(name,n);

}

void show\_data()

{

cout<<name<<" "<<num;

}

};

int main()

{

Sample s(100,"Ajay");

s.show\_data();

}

1. Compile time erro
2. Runtime error
3. **Ajay 100**
4. None of the above

Q4.#include<iostream>

using namespace std;

class A

{

static int c;

public:

A()

{

c++;

}

void show()

{

cout<<c;

}

};

int A :: c=0;

int main()

{

A a,b,c;

a.show();

}

A. 0

**B. 3**

C. 1

D error

Q5.class Test

{

Private:

Test()

{

}

Public:

Test(int x,int y)

{

}

};

Int main()

{

Test t(10,20);

}

A. Program will give compile time error

**B. Program will run fine but without any output.**

C. Program will give runtime error

D. Program will give logical error

Q6.Which among the following is correct?

a.Program will give compile time error

**b. Program will run fine but without any output.**

c. Program will give runtime error

d. Program will give logical error

Q7.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**

e. All of these

Q8. #include<iostream>

#include<cstring>

using namespace std;

class A

{

const int count=20;

public:

A(int x)

{

count=x;

}

void show()

{

cout<<count;

}

};

int main()

{

A a(80);

a.show();

}

A. 20

B.80

**C.compile time error**

D.garbage value

Q9.#include<iostream>

using namespace std;

class A

{

int count;

public:

A(int x=20)

{

count=x;

}

void show()

{

cout<<count;

}

};

int main()

{

A a();

A a1(80);

a.show();

A1.show();

}

**A.20,80**

B.20,20

C.80,80

D.compile time error

Q10.Which of the following statements are not true about destructor?  
  
1. It is invoked when object goes out of the scope  
2. Like constructor, it can also have parameters   
3. It can be virtual   
4. It can be declared in private section  
5. It bears same name as that of the class and precedes Lambda sign.

a. Only 2, 3, 5

b**.** Only 2, 3, 4

**c. Only 2, 4, 5**

d. Only 3, 4, 5

Q11. #include<iostream>

using namespace std;

class destruct

{

int count;

public:

destruct(int i)

{

count=i;

}

~destruct()

{

cout<<"object "<<count<<" destroyed";

}

};

int main()

{

destruct d1(1),d2(2);

}

1. Object 1 destroyed.
2. Object 1 destroyed,object 2 destroyed.
3. **Object 2 destroyed, object 1 destroyed.**
4. Object 2 destroyed.

Q12.Which of the following statements is incorrect?2

a)Friend keyword can be used in the class to allow access to another class.

b) Friend keyword can be used for a function in the public section of a class.

c) Friend keyword can be used for a function in the private section of a class.

**d) Friend keyword can be used on main().**

Q13.

#include<iostream>

using namespace std;

class test

{

int x,y;

public:

void get()

{

x=50;

y=100;

}

friend int compute(test );

};

int compute(test t1)

{

return t1.x+t1.y-5;

}

int main()

{

test t;

t.get();

cout<<compute(t);

return 0;

}

**A. 145.**

B. 150

C. 100

D. 50

Q14. Which of the following is true for a friend function?

1.It is not in the scope of the class to which it has been declared as friend.

2.It cannot be called using the object of the class

3.It can access the private and protected members of the class.

4.If a class A is friend of B, then B becomes friend of A automatically.

A. All of the above.

**B. Only 3,4**

C. 1,2

D. Only 1,2,3.

Q15.

class A

{

Int func();

};

Class B

{ };

What is to be written in class B to declare func() as the friend function to it(class B).

1. Friend int func().
2. **Friend int A :: func();**
3. Friend func();
4. Cannot be declared as friend function.

Q15.Which of the following statements is correct about the program given below?

#include<iostream>  
using namespace std;  
class Test  
{  
    public:  
        static void Mytest();  
};  
int main()  
{  
    void(\*ptr)()=&Test::Mytest;  
    return 0;   
}

A. The program reports an error as pointer to member function cannot be defined outside the definition of class.

B. The program reports an error as pointer to static member function cannot be defined.

C. The program reports an error as pointer to member function cannot be defined without object.

**D. The program reports linker error.**

**Q16.**#include<iostream>  
using namespace std;  
class A  
{  
    static int a;  
    int \*ptr;  
    int b;  
};  
int A :: a=0;  
int main()  
{  
    A obj;  
    cout<<sizeof(obj);  
    cout<<sizeof(A \*);  
}

A. 12 4

**B. 8 4**

C. 12 12

D. 8 8

Q17.#include <iostream>  
    using namespace std;  
    class rectangle  
    {  
        int length, breadth;  
        public:  
        void enter(int, int);  
        int area ()  
        {  
            return (length \* breadth);  
        }  
    };  
    void rectangle::enter (int l, int b)  
    {  
        length = l;  
        breadth = b;  
    }  
    int main ()  
    {  
        rectangle R;  
        R.enter (3, 4);  
        cout << "rect area: " << R.area();  
        return 0;  
    }

Predict the output.

1. **rect area:12**
2. rect area:24
3. Compile time error
4. None of the mentioned.

Q18. Where can the default parameter be placed by the user?

1. Leftmost
2. **Rightmost**
3. Both a & b
4. None of the mentioned.

Q19.Statically created object for class X is:

A.  X \* obj=new X();

**B. X obj;**

C. X obj=new X();

D. None

Q20.#include <iostream>

    using namespace std;

    void func(int a, bool flag = true)

    {

        if (flag == true ) {

            cout << "Flag is true. a = " << a;

     }

        else {

            cout << "Flag is false. a = " << a;

        }

    }

    int main()

    {

        func(200, false);

        return 0;

}

A.Flag is true. a = 200

B.Flag is false. a = 100

**C.Flag is false. a = 200**

D.Flag is true. a = 100