

# Programming in Modern C++: Assignment Week 3

Total Marks : 20

Partha Pratim Das  
Department of Computer Science and Engineering  
Indian Institute of Technology  
Kharagpur – 721302  
partha.p.das@gmail.com

July 31, 2023

## Question 1

Consider the following program.

*[MCQ, Marks 2]*

```
#include <iostream>
#include <string>
using namespace std;
class Sample {
    string name;
public:
    Sample(){
        cout << "s" << " ";
    }
    Sample(string s): name(s) {
        cout << name << " ";
    }
};
int main() {
    Sample s1; //LINE-1
    Sample *s2 = new Sample("s2");
    Sample *s3;
    new Sample("s4");
    return 0;
}
```

What will be the output/error?

- a) Compilation error at LINE-1
- b) s s2 s s4
- c) s s2 s4
- d) s2 s4

**Answer:** c)

**Explanation:**

The statement `Sample s1;` is not an error, but it instantiates an object by calling default

constructor. Hence it prints `s`.

Statement `Sample *s2 = new Sample("s2");`, instantiate an object, and call parametrized constructor. Hence it prints `s2`.

Statement `Sample *s3;`, just create a pointer, don't instantiate an object.

Statement `new Sample("s4");`, creates a temporary object, and call the parametrized constructor. Hence prints `s4`.

Hence, the correct option is c).

## Question 2

Consider the code segment given below.

[MCQ, Marks 2]

```
#include <iostream>
using namespace std;
int i = 0;
class myClass {
    public:
        myClass() { i += 1;}
        ~myClass() { i += 5; }
};
void f(){
    myClass m1();
    myClass m2;
}
int fun() {
    i = 3;
    f();
    return i++;
}
int main() {
    cout << fun() << " ";
    cout << i << endl;
    return 0;
}
```

What will be the output?

- a) 3 4
- b) 15 16
- c) 9 10
- d) 4 5

**Answer:** c)

**Explanation:**

i is initialized to 0 (i = 0; executes before main() is called).

Then main() starts and calls fun() which sets i to 3 by i = 3;. The fun() calls the function f(). In function f(), first myClass m1(); is called, which does not instantiate any object. Hence, this statement does not change the value of i. The next statement calls class constructor, which makes the value of i as 4. As the function f() returns, the destructor of local object m2 will be called before return and i become 9. This 9 will be returned by fun(), and get printed.

Further i value incremented to 10 after return (Since i++ is post incremented).

Finally, value of i is printed as 10.

Hence correct option is c).

### Question 3

Consider the following code segment.

[MSQ, Marks 2]

```
#include <iostream>
using namespace std;
class Private_Data {
    int x;
    void f1() {
        cout << "Within f1";
    }
public:
    int y;
    void f2() {
        cout << "Within f2";
    }
};
int main() {
    Private_Data t;
    t.x = 1;      // LINE-1
    t.f1();       // LINE-2
    t.y = 2;      // LINE-3
    t.f2();       // LINE-4
    return 0;
}
```

Which line/s will generate compilation error/s?

- a) LINE-1
- b) LINE-2
- c) LINE-3
- d) LINE-4

**Answer:** a), b)

**Explanation:**

All the class members declared under **public** accessible inside and outside the class. The class members declared as **private** can be accessed only by the public functions inside the class.

LINE-1 gives error as **x** is **private** member.

LINE-2 gives error as **f1()** is **private** member function.

LINE-3 is fine as **y** is **public** member.

LINE-4 is fine as function **f2()** is **public** member function.

## Question 4

Consider the code segment given below.

[MCQ, Marks 2]

```
#include<iostream>
using namespace std;
class MyClass {
    public:
        MyClass(int i) { cout << i; }
        MyClass(const MyClass &t) { cout << "2"; }
};
int main(){
    MyClass *t1, *t2;          //LINE-1
    t1 = new MyClass(0);       //LINE-2
    t2 = new MyClass(*t1);     //LINE-3
    MyClass t3 = *t1;          //LINE-4
    MyClass t4 = *t2;          //LINE-5
    return 0;
}
```

What will be the output?

- a) 02020202
- b) 0202
- c) 0222
- d) 0002

**Answer:** c)

**Explanation:**

At LINE-1, the statement `MyClass *t1, *t2;` declares two pointers. So no objects are instantiated, hence the constructors are not called.

At LINE-2, the statement `t1 = new MyClass(0);` invokes the parameterized constructor. Hence the output is 0.

At LINE-3, the statement `t2 = new MyClass(*t1);` invokes the copy constructor. Hence the output is 2.

At LINE-4 and 5, the statements `MyClass t3 = *t1;` and `MyClass t4 = *t2;` invoke the copy constructor. Hence the output will be 22.

So option c) is correct.

## Question 5

Consider the code segment.

*[MSQ, Marks 2]*

```
class Check {  
    // code...  
};  
int main() {  
    const Check c; // LINE-1  
    return 0;  
}
```

What is the type of `this` pointer associated with the object `c`?

- a) `const Check* this`
- b) `Check* const this`
- c) `Check const* const this`
- d) `const Check* const this`

**Answer:** c), d)

**Explanation:**

`this` pointer is always a constant. So for class `Check`, the type of `this` for `Check c` would be `Check * const`.

In LINE-1, the object is a constant. So the type of the `this` pointer of a constant object (as specified `const Check`) of class `Check` is:

`const Check* const this`; or `Check const* const this`;

## Question 6

Consider the code segment given below.

*[MCQ, Marks 2]*

```
#include <iostream>
using namespace std;
class Test{
    public:
        Test() { cout << "0" << endl; }
        Test(int x=0) { cout << "K" << endl; }
};

int main(){
    Test t1;
    return 0;
}
```

What will be the output/error?

- a) 0
- b) K
- c) OK
- d) Compilation error: call of overload Test() is ambiguous

**Answer:** d)

**Explanation:**

The program contains an ambiguous definition of `Test` constructor, where the default constructor and parameterized constructor with default parameter make ambiguity in creating objects. Hence, the program will give a compilation error.

## Question 7

Consider the code segment below.

[MSQ, Marks 2]

```
#include <iostream>
using namespace std;
class Point {
    int x, y;
public:
    Point(int _x, int _y) : x(_x), y(_y) { }
    Point(Point &c) : x(c.x), y(c.y){ }
    void change(Point *new_c) { this = new_c; }
    void show() { cout << x << ", " << y << endl; }
};
int main() {
    Point c1(10, 20);
    Point c2(20, 50);
    Point c3(c1);
    c3.change(&c2);
    c3.show();
    return 0;
}
```

What will be the output/error?

- a) 10 20
- b) 20 50
- c) Compilation Error: lvalue required as left operand of assignment
- d) Compilation Error: private data members are inaccessible

**Answer:** c)

**Explanation:** In the function `change(&c2)`, the statement

`this = new_c;`

attempts an assignment to `this`. Since `this` is a constant pointer (`Point * const`), it cannot be changed and the error occurs during compilation.



## Question 8

Consider the code segment given below.

[MCQ, Marks 2]

```
#include <iostream>
#include <string>
using namespace std;
class Data{
    int _d = 0;
public:
    int set_d(int d) const {
        _d = d;
    }
    int get_d() const {
        return _d;
    }
};
int main(){
    const Data obj;
    obj.set_d(5);
    cout<<obj.get_d();
    return 0;
}
```

What will be the output/error?

- a) 0
- b) 5
- c) 6
- d) Compilation error: assignment of member 'Data::\_d' in read-only object

**Answer:** d)

**Explanation:** As the `set_d()` is a constant function, it cannot change the state of an object. Hence, when we try to assign value to `_d` it gives compiler error, i.e. option d).

## Question 9

Consider the code segment given below.

*[MSQ, Marks 2]*

```
#include<iostream>
using namespace std;
class Print {
    int x;
public:
    Print(int _x): x(_x) { }
    void display() { cout << _____ << " "; } //LINE-1
};
int main() {
    Print i(1);
    i.display();
    return 0;
}
```

Fill in the blank at LINE-1 so that the program will print 2.

- a) ++x
- b) ++this->x
- c) x++
- d) ++this.x

**Answer:** a), b)

**Explanation:**

When the `display(.)` function is called, the value of `x` is 1. So, we need to increment `x` before printing. It can be done using `++this->x` or `++x`.

# Programming Questions

## Question 1

Consider the following program. Fill in the blanks as per the instructions given below:

- Complete two constructor statements at LINE-1 and LINE-2,
- Complete the return statement at LINE-3 to calculate the manhattan distance between two points,

such that it will satisfy the given test cases.

*Marks: 3*

```
#include<iostream>
#include<cmath>
using namespace std;
class Point{
    const int x,y;
public:
    Point(int _x=0, int _y=0) : _____ {} //LINE-1
    Point(const Point& p) : _____ {} //LINE-2
    double distance(Point p){
        return _____; //LINE-3
    }
    void print(){ cout << "(" << x << "," << y << ")" << endl; }
};
int main(){
    int x1,x2,y1,y2;
    cin >> x1 >> y1 >> x2 >> y2;
    Point p1(x1,y1), p2(x2,y2);
    p1.print();
    p2.print();
    cout << "Distance: " << p1.distance(p2);
    return 0;
}
```

### Public 1

Input: 1 2 3 4

Output:

(1, 2)

(3, 4)

Distance: 4

### Public 2

Input: 1 -1 2 -2

Output:

(1, -1)

(2, -2)

Distance: 2

## Private

Input: 2 4 6 8

Output:

(2, 4)

(6, 8)

Distance: 8

### Answer:

LINE-1: `x(_x), y(_y)`

LINE-2: `x(p.x), y(p.y)`

LINE-3: `(abs(x-p.x) + abs(y-p.y))`

### Explanation:

The parameterized constructor at LINE-1 will be filled as `x(_x), y(_y)`.

The copy constructor at LINE-2 will be filled as `x(p.x), y(p.y)`.

The manhattan distance between two points can be calculated at LINE-3 as `(abs(x-p.x) + abs(y-p.y))`

## Question 2

Consider the following program. Fill in the blanks as per the instructions given below.

- at LINE-1 with constructor definition,
- at LINE-2 with appropriate statement to deallocate array memory

such that it will satisfy the given test cases.

*Marks: 3*

```
#include<iostream>
using namespace std;
class Array{
    int *arr;
    int size;
public:
    Array(int n) : _____{} //LINE-1
    ~Array(){ _____; } //LINE-2
    void Enter(){
        for(int i=0;i<size;i++)
            cin >> arr[i];
    }
    void FindMin(){
        int min = 999;
        for(int i=0;i<size;i++){
            if(min > arr[i])
                min = arr[i];
        }
        cout << "Min: " << min;
    }
};
int main(){
    int n;
    cin >> n;
    Array a(n);
    a.Enter();
    a.FindMin();
    return 0;
}
```

### Public 1

Input: 4

1 3 5 8

Output: Min: 1

### Public 2

Input: 3

7 3 9

Output: Min: 3

## Private

Input: 5

3 6 4 7 9

Output: Min: 3

## Answer:

LINE-1: `arr(new int[n]), size(n)`

LINE-2: `delete[] arr`

## Explanation:

The parameterized constructor of the class should initialize the data members of the class. Using the constructor, we need to allocate memory to the data member `arr` and initialize `size`. So, LINE-1 needs to be filled as `arr(new int[n]), size(n)`. The destructor should free memory which is allocated dynamically to the data member `arr`. So, LINE-2 should be filled as `delete[] arr`.

### Question 3

Consider the following program. Fill in the blanks at LINE-1, LINE-2 and LINE-3 with appropriate keywords such that it will satisfy the given test cases. *Marks: 3*

```
#include<iostream>
using namespace std;
class Student{
    const int sid;
    string sname;
    _____ int marks; //LINE-1
public:
    Student(int a, string b, int c) : sid(a), sname(b), marks(c) {}
    void updateMarks(int x) _____ { marks += x; } //LINE-2
    void print() _____ { //LINE-3
        cout << sid << " : " << sname << " : " << marks;
    }
};
int main(){
    string n;
    int i, m, u;
    cin >> i >> n >> m >> u;
    const Student s1(i, n, m);
    s1.updateMarks(u);
    s1.print();
    return 0;
}
```

#### Public 1

Input: 1 Raj 36 5  
Output: 1 : Raj : 41

#### Public 2

Input: 2 Amal 48 20  
Output: 2 : Amal : 68

#### Private

Input: 3 John 80 10  
Output: 3 : John : 90

#### Answer:

LINE-1: mutable

LINE-2: const

LINE-3: const

#### Explanation:

The object of class `Student` is declared as constant. So, all functions of class `Student` should be declared as `const` in order to access them using the constant object. So, LINE-2 and LINE-3 will be filled as `const`. The data member marks of constant object is being changed using function `updateMarks()`. Hence, LINE-1 should be filled as `mutable`.