

Programming in Modern C++: Assignment Week 3

Total Marks : 25

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

January 31, 2023

Question 1

Consider the following program.

[MSQ, Marks 2]

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

class myClass{
    int pra = 5;
    public:
        int pub = 10;
        void set_pr(int x){ pra = x; }
        void set_pu(int x){ pub = x; }
};

int main(){
    myClass m;
    int a, b;
    a = m.pra;           //LINE-1
    b = m.pub;           //LINE-2
    m.set_pr(100);       //LINE-3
    m.set_pu(200);       //LINE-4
    return 0;
}
```

Which line/s will generate an error?

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

Answer: a)

Explanation:

At LINE-1, `main()` function is accessing the private data member using object `m` which will generate an error.

At LINE-2, public member is being accessed using object `m` which is fine.

At LINE-3 and LINE-4, public functions are called using object of the class which will not generate any error.

This question is intentionally made as MSQ

Question 2

Consider the following class.

[MCQ, Mark 1]

```
class Test{
    -----:
        int x;
    -----:
        int y;
    /* Some more code */
};
```

Fill in the blanks with proper access specifiers so that member `y` can be accessed from outside of the class but member `x` cannot be accessed.

- a) `public, public`
- b) `public, private`
- c) `private, public`
- d) `private, private`

Answer: c)

Explanation:

A class member should be declared as `public` to be accessed from outside of the class. On the other hand, a private data member prevents itself to be accessed directly from outside of the class.

Question 3

Consider the following code segment.

[MSQ, Marks 2]

```
#include <iostream>
using namespace std;
class myClass{
    public:
        myClass() { cout << "1st" << endl; }
        myClass(int x=0) { cout << "2nd" << endl; }
};

int main(){
    myClass m1;
    return 0;
}
```

What will be the output/error?

- a) 1st
- b) 2nd
- c) 1st
2nd
- d) Compilation Error: call of overloaded 'myClass()' is ambiguous

Answer: d)

Explanation:

The program contains an ambiguous definition of `myClass` 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 4

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
using namespace std;
int i = 0;
class test {
    public:
        test() { i = 1;}
        ~test() { i = 2; }
};
void f(){
    test t;
}
int fun() {
    i = 3;
    f();
    return i++;
}
int main() {
    cout << fun() << " ";
    cout << i << endl;
    return 0;
}
```

What will be the output?

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

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()`, `test t` set `i = 1`. But the object `t` is local within function `f()`.

So, as the function `f()` returns, the destructor of local object `t` will be called before return and `i` become 2. This 2 will be returned by `fun()`, and get printed.

Further `i` value incremented to 3 after return (Since `i++` is post incrementer). Finally, value of `i` is printed as 3.

Hence the correct option is c).

Question 5

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
#include <cstring>
using namespace std;
class Fun {
    char _____; // LINE-1: declare the data members
public:
    Fun(char* _fstr, char* _mstr, char* _lstr) :
        fstr(setFstr(_fstr)), mstr(setMstr(_mstr)),
        lstr(setLstr(_lstr)){}
    char* setFstr(char* fn) {
        cout << fn << " ";
        return strdup(fn);
    }
    char* setMstr(char* mn) {
        cout << mn << " ";
        return strdup(mn);
    }
    char* setLstr(char* ln) {
        cout << ln << " ";
        return strdup(ln);
    }
};
int main() {
    Fun obj("coding", "is", "fun");
    return 0;
}
```

Fill in the blank at LINE-1 so that the program will print
fun is coding.

- a) *lstr, *mstr, *fstr
- b) *mstr, *fstr, *lstr
- c) *lstr, *fstr, *mstr
- d) *fstr, *lstr, *mstr

Answer: a)

Explanation:

The order of invocation to initialization-list function depends on the sequence of the data members declared in the class.

Question 6

Consider the following code segment.

[MCQ, Marks 2]

```
#include<iostream>
using namespace std;
class Point {
    int _x;
    int _y;
    Point(int x, int y) {
        _x = x;
        _y = y;
        cout << _x << " " << _y;
    }
};
int main() {
    Point pt(2, 7);
    return 0;
}
```

What will be the output?

- a) 2 7
- b) 0 0
- c) Compilation error: no default constructor
- d) Compilation error: constructor is private

Answer: d)

Explanation:

The parameterized constructor `Point(int x, int y)` is private by default, so it cannot be accessed by from `main()` function. Hence option d) is correct.

Question 7

Consider the following code segment.

[MCQ, Marks 2]

```
#include <iostream>
using namespace std;
class Complex {
    int x, y;
public:
    Complex(int _x, int _y) : x(_x), y(_y) { }
    Complex(Complex &c) : x(c.x), y(c.y){ }
    void change(Complex *new_c) { this = new_c; }
    void show() { cout << x << ", " << y << endl; }
};

int main() {
    Complex c1(10, 20);
    Complex c2(20, 50);
    Complex 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 (`Complex * const`), it cannot be changed and the error occurs during compilation.

Question 8

Consider the following code segment.

[MCQ, Marks 2]

```
#include<iostream>
using namespace std;
class Test {
    int x;
    public:
        Test(int a=0) : x(a){ cout << "1st "; }
        Test(const int &i) : x(i){ cout << "2nd "; }
};
int main() {
    Test m1=5;
    return 0;
}
```

What will be the output/error?

- a) 1st
- b) 2nd
- c) 1st
2nd
- d) Compilation error: conversion from 'int' to 'Test' is ambiguous

Answer: d)

Explanation:

Both the constructors take an integer as input and assigns it to the data member of the class. So, when the assignment of integer to the class object is done, compiler become confused which constructor to call. Hence, it gives an error.

Question 9

Consider the following code segment.

[MSQ, Marks 2]

```
#include<iostream>
using namespace std;
class constC{
    ----- x; //LINE-1
public:
    constC(int _x) : x(_x) {}
    void setx(int a) const{
        x = a;
    }
    void print() const{
        cout << x << endl;
    }
};
int main(){
    const constC m(1);
    m.setx(5);
    m.print();
    return 0;
}
```

Fill in the blank at LINE-1 such that the program will print 5.

- a) int
- b) const int
- c) mutable int
- d) int mutable

Answer: c), d)

Explanation:

To change the value of the data member of a constant object, we need to declare the data member as mutable. So, the syntaxes are `mutable int` or `int mutable`.

Programming Questions

Question 1

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

3

```
#include<iostream>
using namespace std;
class Employee{
    const int id;
    string name;
    _____ int salary; //LINE-1
public:
    Employee(int a, string b, int c) : id(a), name(b), salary(c) {}
    void updateSalary(int x) _____{ salary += x; } //LINE-2
    void print() _____{ //LINE-3
        cout << id << " : " << name << " : " << salary;
    }
};
int main(){
    string n;
    int i, m, u;
    cin >> i >> n >> m >> u;
    const Employee e1(i, n, m);
    e1.updateSalary(u);
    e1.print();
    return 0;
}
```

Public 1

Input:

1 Ram 1000

3000

Output: 1 : Ram : 4000

Public 2

Input:

2 Rohit 5000

2000

Output: 2 : Rohit : 7000

Private

Input:

3 Rahul 10000

5000

Output: 3 : Rahul : 15000

Answer:

LINE-1: mutable

LINE-2: const

LINE-3: const

Explanation:

The object of class `Employee` is declared as constant. So, all functions of class `Employee` 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 `updateSalary()`. Hence, LINE-1 should be filled as `mutable`.

Question 2

Consider the following program.

- Fill in the blank at LINE-1 with the appropriate initializer statement for the parameterized constructor,
- Fill in the blank at LINE-2 with the appropriate statement which deletes the dynamically allocated memory to data member `arr`

The program must satisfy the sample input and output.

Marks: 3

```
#include<iostream>
using namespace std;
class Array{
    char *arr;
    int size;
public:
    Array(int n) : _____{} //LINE-1
    ~Array(){ _____; } //LINE-2
    void EnterElement(){
        for(int i=0;i<size;i++){
            cin >> arr[i];
        }
    }
    void FindMax(){
        char max = ' ';
        for(int i=0;i<size;i++){
            if(max < arr[i])
                max = arr[i];
        }
        cout << "Max: " << max;
    }
};

int main(){
    int n;
    cin >> n;
    Array a(n);
    a.EnterElement();
    a.FindMax();
    return 0;
}
```

Public 1

Input: 3

a s d

Output: Max: s

Public 2

Input: 5

a e i o u

Output: Max: u

Private

Input: 4

a r u p

Output: Max: u

Answer:

LINE-1: `arr(new char[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 char[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 program below.

- Fill in the blank at LINE-1 to complete parameterized constructor
 - Fill in the blank at LINE-2 to complete copy constructor
 - Fill in the blank at LINE-3 to complete return statement
- The `area()` evaluates the area of the given right-angled triangle.

The program must satisfy the given test cases.

Marks: 3

```
#include <iostream>
using namespace std;
class triangle{
    int *b, *h;
public:
    triangle(int _b, int _h) : b(_____), h(_____){} //LINE-1
    triangle(triangle &t) : b(_____), h(_____){} //LINE-2
    ~triangle(){ delete b; delete h; }
    int area(){ return _____; } //LINE-3
};
int main(){
    int a, b;
    cin >> a >> b;
    triangle *tri = new triangle(a, b);
    cout << tri->area();
    return 0;
}
```

Public 1

Input: 4 7

Output: 14

Public 2

Input: 10 20

Output: 100

Private

Input: 8 4

Output: 16

Answer:

```
LINE-1:  new int(_b), new int(_h)
LINE-2:  new int(*t.b), new int(*t.h)
LINE-3:  0.5 * *b * *h
```

Explanation:

Since the data-members `*b` and `*h` are of pointer type, dynamic allocation (using `new` operator) is required in initialization. The allocation of memory and initialization at **LINE-1** must be done as follows:

```
new int(_b),
new int(_h)
```

At **LINE-2**, we have copy constructor. The allocation of memory and initialization at **LINE-2** must be done as follows:

```
new int(*r.w),
new int(*r.h)
```

The return statement in the function `area()` must be:

```
return 0.5*b * *h;
```