

Programming in C++: Assignment Week 3

Total Marks : 20

August 6, 2017

Question 1

What is the output of the sizeof operator for 't' in the following code snippet ? (Assume sizeof(int) = 4) *Mark 1*

```
#include<iostream>
using namespace std;
class Test {
    int var;
    int arr[9];
    void display() { int a;}
};

int main() {
    Test t;
    cout << sizeof(t) << " ";
}
```

- a) 40
- b) 36
- c) 44
- d) Default size: 0

Answer: a)

Explanation: Sum of memory requirements for all the data members

Question 2

What will be the output of the following program? *Mark 1*

```
#include<iostream>
using namespace std;
class Sample {
    int data_;
    Sample(): data_(2){}
};

int main() {
```

```

    Sample s;
    s.data_ = 1;
    cout << Sample.data_;
}

```

- a) 1
- b) 2
- c) Compilation Error: Sample() is private
- d) 0

Answer: c)

Explanation: Constructor is private, hence object construction not possible

Question 3

What is the output of the program? *Mark 1*

```

#include<iostream>
using namespace std;
class Test {
private:
    int x_;
    int y_;
public:
    void func() {
        x_ = y_ = 1;
        cout << x_ << " " << y_;
    }
};

int main() {
    Test t;
    t.func();
}

```

- a) 1 1
- b) Compilation error: Constructor not defined
- c) Compilation error: Cannot access private member x_ and y_
- d) Compilation error: Illegal access of func()

Answer: a)

Explanation: private members can be accessed in public methods

Question 4

Consider Object S of class **Sample**. What is the type of *this* pointer? *Mark 1*

- a) S const * const this

- b) S * const this
- c) S * this
- d) const S const * this

Answer: b)

Explanation: As per syntax of this pointer

Question 5

What will be the output of the program? *Mark 2*

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

class Sample {
    string name;

public:
    Sample(string s): name(s) {
        cout << name << " Created" << " ";
    }

    ~Sample() {
        cout << name << " Destroyed" << " ";
    }
};

int main() {
    Sample s1("s1"), s2("s2");

    return 0;
}
```

- a) S1 Created S2 Created S2 Destroyed S1 Destroyed
- b) S1 Created S2 Created S1 Destroyed S2 Destroyed
- c) S2 Created S1 Created S2 Destroyed S1 Destroyed
- d) S1 Created S1 Destroyed S2 Created S2 Destroyed

Answer: a)

Explanation: order of calling constructors and destructors, when the object goes out of scope. The last constructed class is destroyed first. See Slides.

I Programming Assignment

Question 1

Fill the blank with the proper constructor and copy constructor to get the output as per the test cases. *Marks 2*

```
#include <iostream>
using namespace std;
class Complex {
    public: double *re, *im;
    Complex(_____) {
        re = new double(r);
        im = new double(m);
    }
    Complex(_____){
        re = new double; im = new double;
        *re = *t.re; *im= *t.im;
    }
    ~Complex(){
        delete re, im;
    }
};

int main() {

    double x, y, z;

    cin >> x >> y >> z;
    Complex n1(x,y);
    cout << *n1.re << "+" << *n1.im << "i ";
    Complex n2 = n1;
    cout << *n2.re << "+" << *n2.im << "i ";
    *n1.im = z;
    cout << *n2.re << "+" << *n2.im << "i ";
    cout << *n1.re << "+" << *n1.im << "i ";
    return 0;
}
```

Answer: double r, double m // const Complex &t

Explanation: The first parameters are for the constructor, the second arguments are for the copy constructor which passes a constant Complex object, so that the value of the data members are not changed.

- a. Input: 4, 5, 6 Output: 4+5i 4+5i 4+5i 4+6i
- b. Input: 4, 5, 5 Output: 4+5i 4+5i 4+5i 4+5i
- c. Input: 6 7 8 Output: 6+7i 6+7i 6+7i 6+8i

Question 2

Fill the blank in the constructor to get the output as per the test cases. *Marks 2*

```
#include <iostream>
using namespace std;
class Sample {
    public:
    int data_ ;
    char graph_, data_or_graph_;
    Sample(_____): data_(x), data_or_graph_(z), graph_(p){
        cout << data_ << " " << data_or_graph_<< " " << graph_ <<" "<<endl;
    }
};
int main() {
    int x; char y;

    cin>>x >> y ;

    Sample s1(x, y), s3;

    return 0;
}
```

Answer: int x = 6, char z = 'C', char p = 'A'// data_(x), data_or_graph_(z), graph_(y)

Explanation: : Evaluation of S3 gives 6 C A hence we get the default values. The rest of the syntax is as per slides.

- a. Input: 4 D Output: 4 D A 6 C A
- b. Input: 71 N Output: 71 N A 6 C A

Question 3

Fill in blank with proper access specifier and function definitions of the class Stack to get the output as per the test cases. *Marks 2*

```
#include <iostream>
#include <vector>
#include<string.h>
using namespace std;
class Stack {
    _____: // Write the appropriate Access specifier
    vector<char> data_; int top_;
    public:
    int empty() { _____; }
    void push(char x) { _____;}
    void pop() { _____; }
    char top() { _____; }
};
int main() {
    Stack s;
```

```

char str[20];

cin >> str;

s.data_.resize(100);
s.top_ = -1;
for(int i = 0; i < strlen(str) ; ++i)
s.push(str[i]);
while (!s.empty()) {
    cout << s.top(); s.pop(); s.pop();
}
return 0;
}

```

Answer: public // return (top_ == -1) // data_[++top_] = x // -top_ // return data_[top_]

Explanation: Access specifier will be public as the data members are accessed outside class. The functions are standard stack functions, refer slides

- a. Input: erty Output: yr
- b. Input: ghjilk Output: kih
- c. Input: ADAM ; Output: MD

Question 4

Look into the main() function write the proper constructor by filling the blank to get the output as per the test cases. *Marks 2*

```

#include <iostream>
#include <cmath>
using namespace std;
class Complex { private: double re_, im_;
public:
    Complex(double re = 4.0, double im = 5.0): re_(re), im_(im)
        { cout << "Ctor: (" << re_ << ", " << im_ << ")" << endl; }
    ~Complex()
        { cout << "Dtor: (" << re_ << ", " << im_ << ")" << endl; }

    void print() { cout << "|" << re_ << "+j" << im_ << "|" << endl; }
};

-----;
int main() {
    cout << "main" << endl;
    double x, y;

    cin >> x;

    cin >> y;
    Complex d(x); Complex e;
    c.print();
    d.print();
    return 0;
}

```

Answer: Complex c(8, 4)

Explanation: Complex object defined before the scope of main

a. Input: 5 6

```
Output
Ctor: (8, 4)
main
Ctor: (5, 5)
Ctor: (4, 5)
|8+j4|
|5+j5|
Dtor: (4, 5)
Dtor: (5, 5)
Dtor: (8, 4)
```

b. Input: 2.5 3.5;

```
Output
Ctor: (8, 4)
main
Ctor: (2.5, 5)
Ctor: (4, 5)
|8+j4|
|2.5+j5|
Dtor: (4, 5)
Dtor: (2.5, 5)
Dtor: (8, 4)
```

Question 5

The program indicates the concept of mutability . Fill the blank with appropriate kew words to satisfy the given test cases *Marks 2*

```
#include <iostream>
using namespace std;
class MyClass {
    int mem_;
    ----- int x_;
public:
    MyClass(int m, int mm) : mem_(m), x_(mm) {}
    int getxMem() ----- { return x_; }
    void setxMem(int i) ----- { x_ = i; }
};
int main() {

    int x, y,z;

    cin >> x;

    cin >> y;
```

```
    cin >> z;  
    const MyClass myConstObj(x, y);  
    myConstObj.setxMem(z);  
    cout << myConstObj.getxMem() << endl;  
    return 0;  
}
```

Answer: mutable // const // const

Explanation: A mutable data member x only can be accessed and updated in a const member function.

- a. Input: 5 7 8 ; Output: 8
- b. Input: 0, 1, 0 ; Output: 0
- c. Input: 11, 11, 11 ; Output: 11