Programming in 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

March 17, 2017

Question 1

What is the output of the sizeof operator? (Assume sizeof(int) = 4) *Mark 1*

```
#include<iostream>
using namespace std;
class Test {
    int x_;
    unsigned char str[8];
};
int main() {
    Test t;
    cout << sizeof(t) << " ";
}
```

a) 10

b) 12

c) Cannot be determined before memory allocation

d) Default size: 0

Answer: b)
Explanation: Sum of memory requirements for all the data members

Question 2

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

```
#include<iostream>
using namespace std;
class Test {
    int x_;
    int y_;
};
int main() {
```

```
    Test.x_ = 1;
    cout << Test.x_;
}
```

a) 1

b) Default value

c) Compilation Error: Cannot modify read only variable

d) 0

Answer: c)
Explanation: Object not created, Invalid access of data member x_ with class name.

Question 3

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

```
#include<iostream>
using namespace std;
class Test {
    int x_;
    int y_;
    void func() {
        x_ = y_ = 1;
        cout << x_ << " " << y_;
    }
};
int main() {
    Test t;
    t.func();
}
```

a) 1 1

b) x = 1 y = 1

c) Compilation error: Cannot access private member

d) None of the above

Answer: c)
Explanation: func() is a private member function, hence cannot be accessed outside class

Question 4

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

a) S * const this

b) S const * const this

c) S * this

d) const S const * this

Answer: a)
Explanation: As per syntax, refer slides

Question 5

Which functions will change the state of the object of class Test? *Mark 1*

```
#include<iostream>
using namespace std;
class Test {
    int x_;
    int y_;
    public:
    void print() { cout << x_ << " " << y_; }
    void setx(int m_) {
        x_ = m_;
    }
    void sety(int n_) {
        y_ = n_;
   }
   int calc1(int n_) {
       int t;
       t = n_ * x_ * y_;
       x_ = n_ * 3;
       return t;
   }
   void calc2(int n_) {
       int t;
       t = n_ * x_;
       cout << t;
   }
};
```

a) Only setx() and sety()

b) Only print()

c) setx(), sety(), calc1(), calc2()

d) setx(), sety(), calc1()

Answer: d)
Explanation: The state of a class is the collection of values of all the member variables of a class at a point. setx(), calc1() and sety() modifies the values of the member variables of the class. Refer slides

Question 6

Consider the program below. An implementation of class Test is shown along with an application section using object of Test. *Mark 1*

```
PROGRAM 1 // Implementation of class Test
#include<iostream>
using namespace std;
class Test {
    int x_;
    int y_;
    public:
    void print() { cout << x_ << " " << y_; }
    void setx(int m_) { x_ = m_;}
    void sety(int n_) { y_ = n_;}
    int calc1(int n_) {
        int t;
        t = n_ * x_ * y_;
        x_ = n_ * 3;
        return t;
    }
    void calc2(int n_) {
        int t;
        t = n_ * x_;
        cout << t;
    }
};

int main() {  // Application section
    Test t;
    t.setx(5);
}
```

Now if we make some changes to the class as given below.

```
PROGRAM 2 // Updated Implementation of class Test
#include<iostream>
using namespace std;
class Test {
    int x_[2];
    int y_;
    public:
    void print() { cout << x_[0] << " " << y_; }
    void setx(int m_) { x_[0] = m_; x_[1] = 0; }
    void sety(int n_) { y_ = n_; }
    int calc1(int n_) {
       int t;
       t = n_ * x_[0] * y_;
       x_[0] = n_ * 3;
       return t;
    }
    void calc2(int n_) {
       int t;
```

```
        t = n_ * x_[0];
        cout << t;
    }
};
```

What changes would be required in the application section?

a) Create different objects

b) No change required

c) Pass different parameters to the setx() and sety() function

d) Pass different parameters to the calc1() and calc2() function

Answer: b)
Explanation: The implementation of the private members of the class is changed, but it did not change the interface of the class. The implementation is not visible outside the class.

Question 7

Consider class `Test`. What are the permissible signatures of a Copy Constructor? *Marks 1*

a) Test(const Test t), Test(Test t);

b) Test(const Test* t), Test(Test* t);

c) Test(Test& t), Test(Test* t);

d) Test(const Test& t), Test(Test& t);

Answer: d)
Explanation: As per syntax, refer slides

Question 8

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

```
#include<iostream>
using namespace std;
class Sample{
    int x;
    int y;
    public:
    void setx(int n) { x = n; }
    void sety(int m) { y = m; }
    int gety() { return y;}
    int getx() { return x; }
};
class Experiment {
    public:
    display(Sample t) { t.setx(8);
    cout << t.x;
    }
};
int main() {
    Sample t;
    Experiment e;
    e.display(t);
}
```

a) 8

b) Compilation Error: setx() method of class Sample cannot be accessed in class Experiment

c) 8 8

d) Compilation Error: Variable x is private in Sample, cannot be accessed in class Experiment

Answer: d)
Explanation: Private data members of a class cannot be accessed by other classes or global functions.

Question 9

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

```
#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 = new Sample("s1");
    Sample * s2 = new Sample("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 s2 Created

Answer: d)
Explanation: s1 and s2 created by new operator, delete operator should be used to call the destructor, as in this case the destructor is not called implicitly.

Question 10

Identify the correct statement(s). *Mark 1*

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

class Employee {
    public:
    string name, addr;
    const int id;
    string dob;

    Employee(string nm, string ad, string dt, int d):
         name(nm), addr(ad), dob(dt), id(d) { }

    void print_attr_dob() const {
        this->dob = "12-02-1986";
        cout << this->dob ;
    }
    void print_attr_name() {
        cout << this->name ;
    }
};

static int count = 1;

int main() {
    const Employee e1("Ram", "Kolkata", "12-02-02", count++);

    e1.print_attr_dob();
    e1.print_attr_name();
    return 0;
}
```

a) Compilation Error: print_attr_dob() cannot assign value to dob

b) Compilation Error: cannot convert 'this' pointer from 'const Employee' to 'Employee &' for print_attr_dob()

c) Compilation Error: cannot convert 'this' pointer from 'const Employee' to 'Employee &' for print_attr_name()

d) B and C

Answer: a), c)
Explanation: A constant object cannot invoke a non constant member function. Constant data member, id cannot be modified in the member function print_attr_dob

I Programming Assignment

Question 1

Write the required syntax for the 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

Write the required syntax for the constructor to get the output as per the test cases. *Marks 2*

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

    cin>>x >> y ;

    Sample s1(x, y), s2(--x, ++y), s3;


    return 0;
}
```

Answer: int x = 5, char z = 'B', int y = 6 // data_(x), data_or_graph_(z), graph_(y)
Explanation: : Evaluation of S3 gives 5, B, 6, hence we get the default values. The rest of the syntax is as per slides.

a. Input: 4 D Output: 4 D 6 3 E 6 5 B 6

b. Input: 3 E Output: 3 E 6 2 F 6 5 B 6

Question 3

Write the required constructor 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) - 1; ++i)
        s.push(str[i]);
        while (!s.empty()) {
            cout << s.top(); 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: ABCDE ; Output: DCBA

b. Input: MADAM ; Output: ADAM

c. Input: APA ; Output: PA

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(_____________________________________): re_(re), im_(im)
         { cout << "Ctor: (" << re_ << ", " << im_ << ")" << endl; }
    ~Complex()
         { cout << "Dtor: (" << re_ << ", " << im_ << ")" << endl; }

     void print() { cout << "|" << re_ << "+j" << im_ << "| " << endl; }
};
Complex c(7.2, 4);
int main() {
     cout << "main" << endl;
     double x, y;

     cin >> x;

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

Answer: double re = 0.0, double im = 0.0
Explanation: The default value of the double parameters of the constructor Complex will be 0.0,0.0 as it is evaluated so in case of `Complex e` call

a. Input: 4.5, 5.5 ;

```
Output:
Ctor: (7.2, 4)
main
Ctor: (4.5, 0)
Ctor: (0, 0)
|7.2+j4|
|4.5+j0|
Dtor: (0, 0)
Dtor: (4.5, 0)
Dtor: (7.2, 4)
```

b. Input: 5.6, 4;

```
Output:
Ctor: (7.2, 4)
main
Ctor: (5.6, 0)
Ctor: (0, 0)
|7.2+j4|
|5.6+j0|
Dtor: (0, 0)
Dtor: (5.6, 0)
Dtor: (7.2, 4)
```

c. Input: 0, 0 ;

```
Output:
Ctor: (7.2, 4)
main
Ctor: (0, 0)
Ctor: (0, 0)
|7.2+j4|
|0+j0|
Dtor: (0, 0)
Dtor: (0, 0)
Dtor: (7.2, 4)
```

Question 5

The program indicates the concept of mutability . Fill the blank with appropriate kew word 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 getMem() const { return mem_; }
    void setMem(int i) { mem_ = i; }
    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: 4, 5, 6 ; Output: 6

b. Input: 1, 1, 0 ; Output: 0

c. Input: 70, 89, 70 ; Output: 70