

Programming in C++: Assignment Week 4

Total Marks : 20

March 22, 2017

Question 1

Using friend operator function, which set of operators can be overloaded? *Mark 1*

- a. \ll , \gg , $<$, $>$, $==$, $<=$, $>=$
- b. $+$, $-$, $/$, $*$
- c. $=$, $()$, $[]$, $->$
- d. $\&\&$, $=$, $*$, $->$

Answer: a

Explanation: As per language syntax, check slides

Question 2

While overloading I/O stream how many number of parameters are required in operator function ? *Mark 1*

- a. 0
- b. 1
- c. 2
- d. 3

Answer: c

Explanation: Check slides

Question 3

What is the output of the following code? *Mark 1*

```
#include <iostream>
using namespace std;
struct emp {
    int a;
    emp ( int b): a(b){}
    ~emp(){ cout << " Destroyed " ;}
    void disp(){ cout << " In Display " ; }
};
int main(){
```

```

    emp e(20);
    cout << e.a ;
    e.disp();
}

```

Output of the Code is

- a. Compilation error
- b. 20 In Display
- c. 20 In Display Destroyed
- d. 0 In Display

Answer: c

Explanation: As per execution semantics of classes and objects, check slides

Question 4

What is the output of the following code? *Mark 1*

```

#include <iostream>
using namespace std ;
namespace Ex { int x = 10; }
namespace Ex { int y = 10; }
int main(){
    using namespace Ex ;
    x = y = 50;
    cout << x << " " << y;
}

```

Output of the Code is

- a. 10 10
- b. 50 50
- c. Error: Cannot Link the namespaces
- d. Compilation error: Invalid Namespace Resolution

Answer: b

Explanation: Using namespace EX to access x and y

Question 5

Fill in the blank. *Mark 1*

```

#include<iostream>
using namespace std;
class Test { static int x;
public:
    void get() { x = 15; }
    void print() {
        x = x + 20;
        cout << "x =" << x << endl;
    }
}

```

```

    }
};
-----; // Define static variable 'x'
int main() {
    Test o1, o2;
    o1.get(); o2.get();
    o1.print(); o2.print();
    return 0;
}

```

- a) `int Test t.x = 0;`
- b) `Test t; t.x = 0;`
- c) `int Test::x = 0;`
- d) `Test t; t::x = 0;`

Answer: c)

Explanation: Static variables are declared and initialised with class name, check slides

Question 6

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

```

#include<iostream>
using namespace std;
class Test { int x;
    public:
    Test(int i) : x(i) {}
    friend void print(const Test& a);
};
void print(const Test& a) {
    cout << "x = " << a.x;
}
int main(){
    Test t(10);
    print(t);
    return 0;
}

```

- a) `x = 10`
- b) Compilation Error: `print` cannot access `x` as it is private
- c) Compilation Error: illegal parameter passing in `print`
- d) Compilation Error: Const parameter cannot be passed in friend function

Answer: a)

Explanation: `x` can be accessed as `print` is a friend function

Question 7

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

```
#include <iostream>
using namespace std;
class sample {
    public:
        int x, y;
        sample() {};
        sample(int, int);
        sample operator + (sample);
};
sample::sample (int a, int b) {
    x = a;
    y = b;
}
sample sample::operator+ (sample param) {
    sample temp;
    temp.x = x + param.x;
    temp.y = y + param.y;
    return (temp);
}
int main () {
    sample a (4,1);
    sample b (3,2);
    sample c;
    c = a + b;
    cout << c.x << " " << c.y;
    return 0;
}
```

- a) 5 5
- b) 7 3
- c) 3 7
- d) 4 6

Answer: b)

Explanation: using operator overloading of + with class Sample objects

Question 8

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

```
#include <iostream>
using namespace std;
class Test {
```

```

    int i;
public:
    Test(int ii) : i(ii) {}
    const Test operator+(const Test& rv) const {
        cout << "Executes +" << endl;
        return Test(i + rv.i);
    }
    Test& operator+=(const Test& rv) {
        cout << "Executes += " << endl;
        i += rv.i;
        return *this;
    }
};

int main() {
    int i = 1, j = 2, k = 3;
    k += i + j;
    Test ii(1), jj(2), kk(3);
    kk += ii + jj;
}

```

- a) Executes +
Executes +=
- b) Executes +
Executes +
- c) Executes +=
Executes +
- d) Compilation Error: Ambiguous declaration

Answer: a)

Explanation: As per precedence

Question 9

Fill in the blanks *Mark 1*

```

#include <iostream>
using namespace std;
class Complex { double re, im; public:
    explicit Complex(double r = 0, double i = 0) : re(r), im(i) { }
    void disp() { cout << re << " +j " << im << endl; }
    friend Complex operator+ (const Complex &a, const Complex &b) {
        return Complex(a.re + b.re, a.im + b.im);
    }
    friend Complex operator+ (const Complex &a, double d) {
        Complex b(d); return a + b;
    }
    ----- {
        Complex a(d); return a + b;
    }
}

```

```

    }
};
int main(){
    Complex d1(2.5, 3.2), d2(1.6, 3.3), d3;
    d3 = d1 + d2; d3.disp();
    d3 = d1 + 6.2; d3.disp();
    d3 = 4.2 + d2; d3.disp();
    return 0;
}

```

- a) friend Complex operator+ (double d, const Complex &a)
- b) friend Complex operator+ (const Complex &b, double d)
- c) friend Complex operator+ (const Complex &a, double d)
- d) friend Complex operator+ (double d, const Complex &b)

Answer: d)

Explanation: As Complex a(d) is created in the body, hence option d

Question 10

Identify the Incorrect statement(s) about static data member of a class. *Mark 1*

- a) It needs to be defined to avoid linker error.
- b) Static data member must be initialized in a source file.
- c) It is Associated with object not with class.
- d) It can be accessed as a member of any object of the class

Answer: c)

Explanation: As per definition of static data members

Programming Assignment

Question 1

Write down the required keywords in the first blank. Fill the rest of the blank by calling the user defined function abs() or library function abs(), So that the given test cases will be satisfied *Marks 2*

```

#include <iostream>
#include <cstdlib>
    ----- NS { // Fill the blank with proper keyword
    int abs(int n) {
        if (n < -128) return 0;
        if (n > 127) return 0;
        if (n < 0) return -n;
        return n;
    }
}
int main() {

```

```

    double x, y, z;
    std::cin >> x >> y >> z ;
    std::cout << _____(x) << " "
    << _____(y) << " "
    << _____(z) << std::endl;
    std::cout <<_____(x) << " "
    << _____(y) << " "
    << _____(z) << std::endl;
    return 0;
}

```

Answer: namespace // NS::abs//NS::abs//NS::abs // abs//abs//abs

Explanation: As per syntax of namespaces, refer slides

a. Input: -203, -69, 9

```

Output: 0    69 9
        203 69 9

```

b. Input: 178, 45, 0

```

Output: 0    45 0
        178 45 0

```

c. Input: -114, 20, 2

```

Output: 114  20  2
        114  20  2

```

Question 2

Here S and R Represent two geometric class, Square and Rectangle respectively. Our objective is to convert /Interpret the Square object as Rectangle and calculating the area of rectangle.

Marks 2

```

#include <iostream>
using namespace std;

class S;

class R {
    int width, height;
public:
    int area ()    // Area of rectangle
    {return (width * height);}
    void convert (S a);
};

class S {

```

```

        _____; // Fill the blank
private:
    int side;
public:
    S (int a) : side(a) {}
};

void _____ (S a) {
    width = a.side;
    height = a.side; // Interpreting Square as an rectangle
}

int main () {

    int x;

    cin >> x;
    R rect;
    S sqr (x);
    rect.convert(sqr);
    cout << rect.area();
    return 0;
}

```

Answer: friend class R// R::convert

Explanation: If a class needs to access the private members(width and height) of a different class, it should be declared as a friend class.

a. Input: 4

Output: 16

b. Input: -6

Output: 36

c. Input: -2.5

Output: 4

Question 3

This Program is all about the implementation of Pre/Post Incrementer. Fill the blank By keeping this in mind so that the given test cases will satisfy. *Marks 2*

```

#include <iostream>
using namespace std;

```



```

class MyClass { int data;
    public:
        -----{ } // Define Constructor
        MyClass& operator++() {
            ++data;
            return -----;
        }
        ----- {
        MyClass t(data);
            ++data;
            return -----;
        }
        void disp() { cout << " " << data ; }
};
int main() {

    int x;

    cin >> x;
    MyClass obj1(x);
    obj1.disp();
    MyClass obj2 = obj1++;
    obj2.disp();
    obj2 = ++obj1;
    obj2.disp();
    return 0;
}

```

Answer: MyClass(int d): data(d) // *this // MyClass operator++(int) // t //

Explanation: As per operational semantics of the post and pre increment operators, check slides.

a. Input: 4

Output: 4 4 6

b. Input: -9

Output: -9 -9 -7

c. Input: 0

Output: 0 0 2

Question 4

Here display() is a non member function which should display the data member of Myclass. Apply the proper concept to fill the blank so that the given test cases will pass. *Marks 2*

```

#include<iostream>
using namespace std;
class MyClass { int x_;
    public:
    MyClass(int i) : x_(i) {}
    -----; // Declare the display function.
};
void display(-----) {
    cout << " " << a.x_;
}
int main(){

    int x;

    cin >> x;
    MyClass obj(x);
    display(obj);
    return 0;
}

```

Answer: friend void display(const MyClass &a); // const MyClass &a

Explanation: To access the private member of a class, a non member function(in this case display) should be declared as a friend function. Check the slides

a. Input: 4

Output: 4

b. Input: 8.7

Output: 8

c. Input: 0

Output: 0

Question 5

Fill the blank by keeping in mind that, the program tests the conceptual knowledge about static

Marks 2

```

#include<iostream>
using namespace std;
class MyClass { static int x;
    public:

```

```

        void get() { x++; }
        ----- print(int y) { //Fill the blank with proper key words
            x = x - y;
            cout << " " << x ;
        }
};
-----; // Define static data member
int main() {

    int x;

    cin >> x;
    MyClass:: print(x);
    MyClass o1;
    o1.get();
    o1.print(x);
    return 0;
}

```

Answer: static void // int MyClass::x = 1

Explanation: Static variables can be initialised outside the scope of the main without constructing objects. It remains live outside main.

a. Input: 5

Output: -4 -8

b. Input: 0

Output: 1 2

c. Input: -7

Output: 8 16