

Assignment (Theory) 2 (Basic Classes and Scopes, 2 points)

Asif Altaf

a) Shortly describe the following terms: member variable, member function, constructor, destructor, default argument, public, private.

Member variable:

A variable that is a part of a specific object, and can be accessed by all of that object's functions.

Member function:

A function that is either declared or defined within the class definition. It has access to all the member variables of that class for the object that it is part of.

Constructor:

A constructor is a special member function of a class that is executed whenever we create new objects of that class. It has the same name as the class and does not have any return type, not even void.

Destructor:

A destructor is a special member function of a class that is executed whenever an object of its class goes out of scope or whenever the delete expression is applied to a pointer to the object of that class. A destructor will have exact same name as the class prefixed with a tilde (~) and it can neither return a value nor can it take any parameters.

Default argument:

A default argument is the value given to a parameter in function declaration. This value is assigned as an argument value in case if programmer deliberately does not pass the value.

Public:

This is an access modifier. A public member of a class is accessible from anywhere outside the class but within a program.

Private:

A private member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members. By default all the members of a class would be private.

Assignment (Theory) 2 (Basic Classes and Scopes, 2 points)

Asif Altaf

b) What is the output of the following code ? Please explain !

Output:

```
construct A: 4
construct A: 2
3 times left
construct A: 1
2 times left
destruct A: -1
1 times left
destruct A: 2
destruct A: 1
```

Explanation:

```
class A {
public:
    int counter;
    A(int a = 1) {
        counter = a;
        cout << "construct A: " << counter << endl;
    }
    ~A() {
        cout << "destruct A: " << counter << endl;
    }
    void use() {
        if( -- counter > 0 ) {
            cout << counter << " times left" << endl;
        }
    }
};

int main()
{
    A a1(4), a2(2);    // constructor is called for each of these object creations with
                       // values 4 and 2 respectively
    a1.use();          // use is called and "3 times left" is printed
    {
        A a3;          // a3 is created inside this scope and constructor is called
                       // for that object with default value 1
        a3.use();       // cout does not run if --counter<=0
        a1.use();       // printed "2 times left"
        a3.use();       // again nothing to print
    }                  // a3's scope ends so destructor is called printing "destruct A:
                       // -1"
    a1.use();          // printed "1 times left"
    return 0;
}                     // scope ends for a1 and a2 so they are destroyed
```