OOPS Lab

Programming Assignment № 2

Constructors and Destructors

Saif Ali, CSE Department, JMI 22/08/2023

Read carefully before you begin:

- Total Marks: 30. Each question carries 10 marks.
- You have 2 hours to complete the assignment. Failure to have your program evaluated before you leave the lab will cause forfeiture of the grade for that lab.
- In order to receive full marks, you must demo the full working code and show the output and given an explanation of your approach where applicable.
- Please **save your code** throughout the semester in a place where you do not lose it. You will be required to submit it at the end.
- Use proper filenaming conventions and commenting. Code that is hard to read or understand will incur a penalty.
- Collaboration must kept to general discussions only. Please do NOT share code or directly share answers with each other. Plagiarism is unacceptable.

Problem 1 (10 marks)

The Class Constructor

A class constructor is a specialized member function of a class. The distinctive property of a constructor is that it is *always* executed whenever we create new objects of *that* class.

A constructor has exactly the same name as the class and it does not have any return type, not even void. Constructors can be very useful for setting initial values for certain member variables.

Examine the following code listing. Identify the class "constructor" in the code. Reproduce this code in your own environment. Build and run and examine the output.

Listing 1: Sample C++ code – a simple C++ class with a constructor. (Source: GeeskforGeeks)

```
1 #include <iostream>
2
3 using namespace std;
4
5 class Line {
```

```
6     public:
7        void setLength( double len );
8        double getLength( void );
9        Line();  // This is the constructor
10    private:
11       double length;
12 };
13
14 // Member functions definitions including constructor
15 Line::Line(void) {
16    cout << "Object is being created" << endl;
17 }
18 void Line::setLength( double len ) {
19    length = len;
20 }
21 double Line::getLength( void ) {
22    return length;
23 }
24
25 // Main function for the program
26 int main() {
27    Line line;
28
29    // set line length
30    line.setLength(6.0);
31    cout << "Length of line : " << line.getLength() <<endl;
32
33    return 0;
34 }
```

Problem 2 (10 marks)

1. Now modify the constructor code so that it sets the length of the line to 0. What effect will this change have on created objects, if any?

2. Now modify the constructor code so that it accepts one parameter called **double inLength**. This is called a "parameterized constructor" because it takes a parameter. Modify the code so that the constructor sets the length to the input parameter.

Problem 3 (10 marks)

A destructor is a special member function of a class that is executed whenever an object of it's class goes out of scope or whenever the delete expression is applied to a pointer to the object of that class.

Examine the code below, identify the destructor. What do you notice about it? Anything special? Build and run the code and examine the destructor. What could the utility of such a feature be for programmers?

Listing 2: Sample C++ code – a simple class definition with a destructor. (Source: GeeksforGeeks)

```
#include <iostream>

using namespace std;
class Line {
   public:
      void setLength( double len );
      double getLength( void );
      Line();   // This is the constructor declaration
      ~Line();  // This is the destructor: declaration

   private:
      double length;
};

// Member functions definitions including constructor
Line::Line(void) {
   cout << "Object is being created" << endl;
}
Line::~Line(void) {
   cout << "Object is being deleted" << endl;
}
void Line::setLength( double len ) {
   length = len;
}
double Line::getLength( void ) {
   return length;
}

// Main function for the program
int main() {
   Line line;

   // set line length
   line.setLength(6.0);
   cout << "Length of line : " << line.getLength() <<endl;

   return 0;
}
```