SLIIT
Discover Your Future

BSc (Hons) in Information Technology
Year 1

Tutorial 07

IT1050 – Object Oriented Concepts **Semester 2, 2021**

Objectives : Learn to create constructors, destructors and dynamic objects

Use your GitHub Repo and Repl.IT account and use the Instructions provided by your Instructors to complete the Tutorial. This week we will implement constructors, destructors and dynamic objects. Use your Repl.IT account and use the Instructions provided by your Instructors to complete the Tutorial. All instructions are in the Repl.IT and GitHub Classrooms for the Tutorial Questions for Week 08. Please submit your solutions using Repl.IT itself.

Exercise 0 – Setting up your Repo

Please see previous tutorials for instructions on setting up your GitHub repo and how to access your code using Repl.IT

Login to your GitHub account and use the following code to clone Tutorial 03 to your repo
https://classroom.github.com/a/M3OGujHb

Students having an issue in connecting to Repl.IT from their GitHub Repo kindly read the instructions provided at the end of the Tutorial 07 sheet.

Exercise 1 – Item Class - Constructors and Destructors

a) In the `Item.h` header file

1. Write the prototype of the default constructor
2. Write the prototype of a overloaded constructor (see main program to identify the parameters)
3. Write the prototype of the destructor

```
class Item {
   private:
      int itemCode;
      float unitPrice;
      float discount; // out of 100 e.g. discount = 15
   public:
      // 1. Write the Default Constructor
      // 2. Write the Overloaded Constructor
      // 3. Write the Destructor

      void setDiscount(float punitPrice);
      float getDiscount();
      float discountedPrice();
      void display();
};
```

b) In `Item.cpp` program

4. Implement the default constructor (initialize all properties to zero)
5. Implement the overloaded constructor
6. Implement the destructor (You should print "Destructor Called")

```
#include "Item.h"
#include <iostream>
using namespace std;

// 4. Implement Default Constructor Implementation
// 5. Implement Overloaded Constructor Implementation
// 6. Implement Destructor (display "Destructor Called")

void Item::setDiscount(float pdiscount) {
  discount = pdiscount;
}

float Item::getDiscount() {
  return discount;
}

float Item::discountedPrice() {
   return unitPrice - unitPrice * discount/100;
}

void Item::display() {
  cout << "Item : " << itemCode << endl;
  cout << "Discounted Price " << discountedPrice() << endl;
}
```

Run the program you should get the following output if your program is correct

```
Item : 0
Discounted Price 0
Item : 100
Discounted Price 800
Destructor Called
Destructor Called
```

Exercise 2 – Shape Classes - Dynamic Objects

- The `Rectangle` class is implemented in `Rectangle.h` and `Rectangle.cpp`
- The `Circle` class is implemented in `Circle.h` and `Circle.cpp`
- You should not change the code in the `Rectangle` and `Circle` classes
- In the `Exercise02.cpp` program create two dynamic objects as instructed below.

1. Create a dynamic `Rectangle` type variable (pointer)
2. Create a dynamic `Rectangle` Object set the length and width that was input from the keyboard
3. Create a dynamic `Circle` type variable (pointer)
4. Create a dynamic `Circle` Object set radius that was input from the keyboard
5. Call the display method of the `Rectangle` Object
6. Call the display method of the `Circle` Object
7. Delete the `Rectangle` Object from memory
8. Delete the `Circle` Object from memory

Do not change any other code in the `Exercise02.cpp`

```
#include "Rectangle.h"
#include "Circle.h"
#include <iostream>
using namespace std;

int main() {
  // ========= DO NOT CHANGE THE INPUT CODE BELOW ==================
   int length, width, radius;
   cout << "Enter length of Rectangle : ";
   cin >> length;
   cout << "Enter width of Rectangle : ";
   cin >> width;
   cout << "Enter radius of Circle : ";
   cin >> radius;
   // ======== DO NOT CHANGE THE CODE GIVEN ABOVE ====================

   // 1. Create a dynamic Rectangle type variable (pointer)
   // 2. Create a dynamic Rectangle Object set the length and width that was input from the keyboard
   // 3. Create a dynamic Circle type variable (pointer)
   // 4. Create a dynamic Circle Object set radius that was input from the keyboard
   // 5. Call the display method of the Rectangle Object
   // 6. Call the display method of the Circle Object
   // 7. Delete the Rectangle Object from memory
   // 8. Delete the Circle Object from memory


   // ============ DO NOT CHANGE THE CODE BELOW ==================
   cout << "End of Program" << endl;

}
```

Finishing Up Tutorial

Use the Repl.IT version icon to commit your code. Goto GitHub and check if your code has been committed.

Exercise 0 (contd)

Students who have issues in connecting to Repl.IT when following instructions given in Exercise 0 in Page 1 can use the following link https://classroom.github.com/a/xhk2oQC2 (please use this only if you have a problem with the first link)