Rajarata University of Sri Lanka

Faculty of Applied Sciences

B.Sc in Information and Communication Technology

ICT 1306 (Object Oriented Programming)
Practical 05

Outline

- Static
- Constants
- Operator Overloading
- Data Conversion

Outcome

At the end of this session students should be able to:

- Get knowledge about the Operator Overloading, Constants and Data Conversion.

---

1. Create a Book class with int type counter, float type price and string type name data members. Initialize counter to 0, price to 0.0 and name to "OOP"

   Whenever a book is created the counter value should increase by 1(increment the counter inside constructor). Include a display function to print all the data members.

   Create a book type object named "b1" and display the values.
   Create another Book named "b2" and display the values.

2. Extend the above programme by adding an int type "pid". Which should contain publisher id.

   Create a constructor to submit values when creating the objects through arguments.
   Inside display function assign value 90 to pid;

   Display values.

   Now make the pid a constant and compile (what is the output???)
   Make the display function a constant function.(What is the output???)

   Create another display2 function with one argument. You should pass a Book object to it. And add 3 to its publisher ID.

   Change the argument to a constant type. (what is the output???)

   Create constant book object and call display and display2 member functions. (What is the output???)

3. Assume you want to increment the counter value in book object. Try the following

   ++object → (what is the output???)

   Insert the following code to the class book

```
void operator ++(){
        ++counter;
```

}

Now try to do the above. (What is the output???)

Try to assign this value to a new book object. (What is the output???). Now fix that issue.

4. Create a class named “Distance”. With feet of int type and inches of float type. Define a constant of type float to store a Meter to Feet conversion value(MTF = 3.28)

Using constructor initialize feet and inches values to 0 and MTF to 3.28

Using another constructor you need to convert meter value to Distance type. (pass the meter as an argument) Conversion is as follows:

```
Float f= MTF*(meter value)

Feet = int(f);

Inches = 12 * (f-feet)
```

Write a showdist() function which is a constant function to view feet and inches values.

Use a conversion operator to convert Distance to a meter value.

```
Operator float() const{

float f=inches/12;

f+=static_cast<float>(feet);

return f/MTF;

}
```

Use main class to test Diatance to meter conversion and meter to Distance conversion.

Next Practical: Inheritance