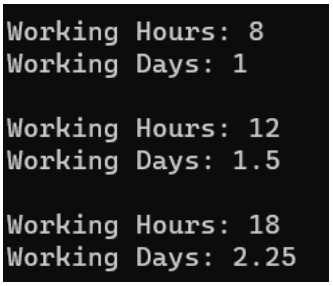
Object-oriented Programming – C++

Tutorial 5

1. Create a class called NumDays. The class aims to store a value representing the number of work hours and convert it to the number of days. For example, 8 hours would be converted to 1 day, 12 hours would be converted to 1.5 days, and 18 hours would be converted to 2.25 days. The class should have a constructor that accepts hour, as well as member functions for retrieving the hours and days. *You must create a private member function to convert the hours to working days.* Test your class by creating three objects from the class.



Sample Output

2. Create a class called Complex for performing arithmetic with complex numbers. Complex numbers have the form

   realPart + imaginaryPart * i

   The class should have two private member variables, *real* and *imag*. Provide a constructor with no arguments that will set a default value for each of the private member variables. Provide another constructor accepting two arguments to initialize the two member variables. Provide public member functions that perform the following tasks:

   a) Adding two Complex numbers: The real parts are added together, and the imaginary parts are added together.
   b) Subtracting two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.
   c) Printing Complex numbers in the form (a, b), where a is the real part and b is the imaginary part.

   Test your class by creating three Complex objects.

3. Write a Circle class that has the following member variables:
    - **radius**: a double
    - **area**: a double
    - **diameter**: a double
    - **circumference**: a double
    - **pi**: a double initialized with the value 3.14159

    The class should have the following **public** member functions:
    - **Default Constructor**. A default constructor that sets radius to 0.0.
    - **Constructor**. Accepts the radius of the circle as an argument.
    - **setRadius**. A mutator function for the radius variable.
    - **getRadius**. An accessor function for the radius variable.
    - **getArea**. Returns the area of the circle.
    - **getDiameter**. Returns the diameter of the circle.
    - **getCircumference**. Returns the circumference of the circle.

    The class should have the following **private** member functions:

    - **calcArea**. Calculate the area of a circle.
    - **calcDiameter**. Calculate the diameter of the circle.
    - **calcCircumference**. Calculate the circumference of the circle.

    In the main function demonstrate the Circle class by creating two Circle objects and then reporting the circle's area, diameter, and circumference.

```
Circle 1
Area: 38.4845
Diameter: 7
Circumference: 21.9911

Circle 2
Area: 84.9486
Diameter: 10.4
Circumference: 32.6725
```

Sample Output

4. In each of the following C++ code that implements static, explain what is wrong with the code.

Sample A:

```
#include <iostream>
using namespace std;

class Sample {
   static int count;

public:
   static Sample()
   {
      count++;
   }
   static void printTotalObj()
   {
      cout << count << endl;
   }
};

int Sample::count = 0;

int main()
{
   Sample S1;
   Sample S2;

   Sample::printTotalObj();
   return 0;
}
```

Sample B:

```
#include <iostream>
using namespace std;

static class Sample {
   static int count;

public:
   Sample()
   {
      count++;
   }
   static void printTotalObj()
   {
      cout << count << endl;
   }
};

int Sample::count = 0;

int main()
{
   Sample S1;
   Sample S2;

   Sample::printTotalObj();
   return 0;
}
```

Sample C:

```
#include <iostream>
using namespace std;

class Sample {
   static int count;

public:
   Sample()
   {
      count++;
   }

   void sayHello()
   {
      cout << "Hello, ";
   }
   static void printTotalObj()
   {
      sayHello();
      cout << count << endl;
   }
};

int Sample::count = 0;

int main()
{
   Sample S1;
   Sample S2;

   Sample::printTotalObj();
   return 0;
}
```

Sample D:

```
#include <iostream>
using namespace std;

class Sample {
   static int count;

public:
   Sample()
   {
      count++;
   }
   void printTotalObj()
   {
      cout << count << endl;
   }
};

int Sample::count = 0;

int main()
{
   Sample S1;
   Sample S2;

   Sample::printTotalObj();
   return 0;
}
```