Digital Skills Academy

FUNDAMENTALS OF PROGRAMMING

LAB 5 – PROBLEM SHEET

http://www.projectcartoon.com/cartoon/2

Conor O'Reilly

Development Approach

- **STAGE 1: Before writing code your code design should be documented in the eLabbook as follows:**
- Problem Definition
  - What is the objective
  - What is the program to do
- Design
  - Draw a picture of the execution steps
  - Write down in words the execution steps
  - Draw the design Object or Class diagrams
- Test Cases (how will you test it)
  - Write what you will use for testing that it runs and creates the right answer
    - Test Case 10 : 1 + 1 = 2 — Simple Case
    - Test Case 20: 5 + 9 = 14 — Normal Case
    - Test Case 30: 0 + 9 = 9 — Edge condition
    - Test Case 40 : 5 + 0 = 5 — Edge condition
- *If this was division we could have division by zero issues and very small answers*
    - Test Case 50 : 166666666666 + 788777777777777 = 788944444444443 — test the very big
- **STAGE 2: Once you have documented your approach you should proceed to do the following:**
- Write Code
  - Step by step, on piece of functionality at a time, get it working, save a copy of that working version ***addTwoNumbers_v1.code*** in your _Attic directory, add the next bit of functionality
- Test Code with test cases
  - Debug the code, change ONLY ONE thing at a time, KEEP SAVING VERSIONS
- **STAGE 3: The code once written should be documented in the eLabBook under the following headings:**
- Code
  - Insert the code into the eLabBook
- Screens
  - Take snapshots of the program screens and copy them into the eLabBook
- Test Records
  - Records of the tests you performed and the results
- documentation
  - How to use the program documentation
  - Object and class diagrams showing the implementation
- References
  - Any websites or code you looked up or used in the creation of your program

PROBLEM 5.1 – AREA OF A TRIANGLE

Similar to the App and Sphere classes shown on the next slides. Write an application to calculate the area of a Triangle, where the area calculation is performed in a Triangle class not in the App class

The area of a triangle is = half the base * the perpendicular height

Objectives

- To create the default App class with main() only instantiating the App class
- To create an instantiable class

App

App.java

```
/*************************************************************************
 *  calculate the volume of a sphere given the radius
 *  using a sphere class
 *
 *  Date: 01 Oct 2011
 *  @author Conor O Reilly
 *
 *************************************************************************/

import javabook.*;

class App
{

    public static void main(String args[])
    {
        App thisProgram = new App();
    }

    public App()
    {
        //Declare variables
        double aRadius;
        double aVolume;

        //Declare objects
        MainWindow mWindow;
        InputBox iBox;
        OutputBox oBox;

        Sphere aSphere;

        //Create objects
        mWindow = new MainWindow();
        iBox = new InputBox(mWindow);
        oBox = new OutputBox(mWindow);

        aSphere = new Sphere();

        //Use objects
        mWindow.show();

        //get Input
        aRadius = iBox.getDouble("Please enter the radius of the sphere: ");
        aSphere.setTheRadius(aRadius);

        //debug code
        System.out.println( "value input:" + aRadius);
        System.out.println( "the radius value in the sphere object:" + aSphere.getTheRadius() );

        //Process
        aSphere.computeVolume();
        aVolume = aSphere.getTheVolume();

        //Output
        oBox.show();
        oBox.print(" The volume of a sphere with a radius of  " + aRadius + " is "+ aVolume );

    }

}
```

Sphere Class

Sphere.java | App.java

```
 6   **********************************************************************/
 7
 8  class Sphere
 9  {
10      // DATA
11      //.....................................................................
12      //Private Constants
13          private final double PI = Math.PI;
14          private final double RATIO_4_OVER_3 = (float)  4 / 3 ;
15
16      //Private Variables
17          private double theRadius;
18          private double theVolume;
19
20      // CONSTRUCTORS
21      //.....................................................................
22      public Sphere()             // same name as the class and the file name
23      {
24          this.theRadius=0;
25          this.theVolume=0;
26      }
27
28      // METHODS - behaviours
29      //.....................................................................
30      public void computeVolume()
31      {
32          this.theVolume = RATIO_4_OVER_3 *  PI * ( this.theRadius * this.theRadius * this.theRadius);
33
34          // using the Math library the formula would be written
35          // this.theVolume = RATIO_4_OVER_3 * Math.PI * Math.pow( this.theRadius, 3);
36      }
37
38      // METHODS - gets (accessors)  and sets (mutators)
39      //.....................................................................
40      public void setTheRadius(double radius)
41      {
42            this.theRadius= radius;
43      }
44
45      public double getTheRadius()
46      {
47            return(this.theRadius);
48      }
49
50      // SHOULD NOT HAVE THIS METHOD, as theVolume is set by computeVolume()
51      // just an example of the use of private - so external classes cannot see this method
52      // and we wouldn't want it set any other way but by the computeVolume method.
53      // Even though it is private it would be better not to have this method at all.
54      private void setTheVolume(double volume)
55      {
56            this.theVolume = volume;
57      }
58
59
60      public double getTheVolume()
61      {
62            return(this.theVolume);
63      }
64
65  }
66
```


Conor O'Reilly

PROBLEM 5.2

Similar to the App and Sphere classes shown on in 4.1 design and write an App and Dice class. On execution the App class should create three Dice objects and toss each one. The results of the toss should be displayed in an output window

The dice class design is given on the next slide

Objectives

- To create the default App class with main() only instantiating the App class
- To create an instantiable class
- To create multiple objects of a class

The dice class

Dice.java

```java
/**********************************************************************
 *
 *  Dice
 *
 *  Date: 14 Oct 2011
 *  @author Conor O Reilly
 *
 *  Maths.Random() retunrns a positive number between 0 and 1
 *  to get a number in a range that you want e.g. 1 to 6 for a dice
 *  you multiply the value produced by the random method by your upper bound e.g. 6
 *  and add on the value of your lower bound e.g. 1 inthis case
 *
 **********************************************************************/

class Dice
{
    // DATA
    //.....................................................
    //Private Constants
    final int NUMBER_OF_SIDES = 6;

    //Private Variables
    private int faceValue;


    // CONSTRUCTORS
    //.....................................................
    public Dice()
    {
        this.faceValue = 0;     //zero if not thrown
    }

    // METHODS - behaviours
    //.....................................................
    public void throwDice()
    {
        this.faceValue = 1 + (int) (Math.random() * NUMBER_OF_SIDES);
    }


    // METHODS - gets (accessors)  and sets (mutators)
    //.....................................................
    public int getFaceValue()
    {
          return(this.faceValue);
    }

}

```

PROBLEM 5.3

Based on the code on 4.2 ask the user if they want to throw one dice, two dice or three dice based on the answer roll the required number of dice and return the results.

Hint: Use IF based on the number of dice input

Objectives

- To create the default App class with main() only instantiating the App class
- To create an instantiable class
- To create multiple objects of a class only when required, so declare but no new is some cases
- Implementation of an if ... else if ... else statement

Conor O'Reilly

PROBLEM 5.4

Create a class that takes as input the test score a student obtained and outputs their grade.

Test Score	Grade
90 ≤ score	A
80 ≤ score < 90	B
70 ≤ score < 80	C
60 ≤ score < 70	D
score < 60	F

```
if (score >= 90)
   mBox.show("Your grade is A");

else if (score >= 80)
   mBox.show("Your grade is B");

else if (score >= 70)
   mBox.show("Your grade is C");

else if (score >= 60)
   mBox.show("Your grade is D");

else
   mBox.show("Your grade is F");
```

Objectives

- To create the default App class with main() only instantiating the App class
- To create an instantiable class
- To create multiple objects of a class only when required, so declare but no new is some cases
- Implementation of an if ... else if ... else statement

Conor O'Reilly

PROBLEM 5.5

Create a leap year class that takes as input the a year.

Outputting if the date is a leap year or not with the rules that were applied.

Leap year rules are:

Is the year evenly divisible by 4? If so, it is a leap year,
unless…
Is the year evenly divisible by 100? (for example, 1500?) If so, it is not a leap year,
unless…
Is the year evenly divisible by 400? If so, it is a leap year.

Objectives

- To create the default App class with main() only instantiating the App class
- To create an instantiable class
- To create multiple objects of a class only when required, so declare but no new is some cases
- Implementation of an if ... else if ... else statement
- Ref : http://support.microsoft.com/kb/214019/EN-US