INTPROG TB2: OBJECT ORIENTED PROGRAMMING

Practical Worksheet 5: Using Classes and Libraries in Java (NetBeans)

1. Introduction

Work through this practical session at your own speed. Make sure you understand everything included within the example before you move onto the questions at the end of the worksheet. If you have questions, either ask your practical tutor, or ask a question using the anonymous question form on Moodle.

2. Example – Stick Figure

We are going to work through an example which involves drawing a stick figure. It will be gradually built on, but we are going to go through step-by-step. This will show you an easy way to gradually build up your program, rather than expecting it to do everything from the beginning.

> Scenario
>
> You have been asked to create a stick figure. The user will need to enter the size of the window into the console – you can assume that the window will be square, so only one dimension will be required. The program will then randomly choose which colour the stick figure will be drawn in from red, blue, and green.

We are going to create the stick figure for this scenario by carrying out the following stages:

1. Create a stick figure of a fixed size with a single colour
2. Adapt the stick figure to be a randomly generated colour
3. Adapt the stick figure to be scalable
4. Get the text entry to decide the window size

Firstly, we need to be able to create our stick figure as a class. We are going to start with a 500 x 500 window in which to draw the stick figure. Also, we will just use the colour red initially. This will test that we are drawing our stick figure correctly, and in proportion.

Create a new project called "StickFigureExample".

Copy the Canvas and KeyboardInput files which are available on Moodle into the project (make sure you select "Refactor Copy" in the paste menu so that the correct package name is entered in the files).

Create a new class called "StickFigure". This class will include your code which will draw the stick figure.

With the exception for the package declaration line (at the top), delete the code which has been included as default, and enter the following:

```
import java.awt.*; // allows us to use the Color class to define the stick figure's colour

/**
* Class to define the drawing of a stick figure in the middle of the provided window
*/
public class StickFigure
{
    private Canvas window;

    /**
    * Constructor for StickFigure
    * @param win The window to draw the stick figure on
    */
    public StickFigure(Canvas win)
    {
        window = win;
    }

    /**
    * Method to draw the stick figure
    */
    public void drawStickFigure()
    {
        // set colour to red (initially for first stage)
        window.setForegroundColor(Color.RED);

        // head
        window.fillCircle(250, 100, 100);

        // body
        window.drawLine(250, 150, 250, 300);

        // arms
        window.drawLine(150, 200, 350, 200);

        // left leg
        window.drawLine(250, 300, 200, 400);

        // right leg
        window.drawLine(250, 300, 300, 400);
    }

}
```

Now that we have the StickFigure class coded, we need to adapt our main method so that it creates a window and then creates a StickFigure object. Enter the following code in your main method (contained within the StickFigureExample file).

```
Canvas stickFigureCanvas = new Canvas("Stick Figure", 500, 500);
StickFigure stickFigure = new StickFigure(stickFigureCanvas);
stickFigure.drawStickFigure();
```

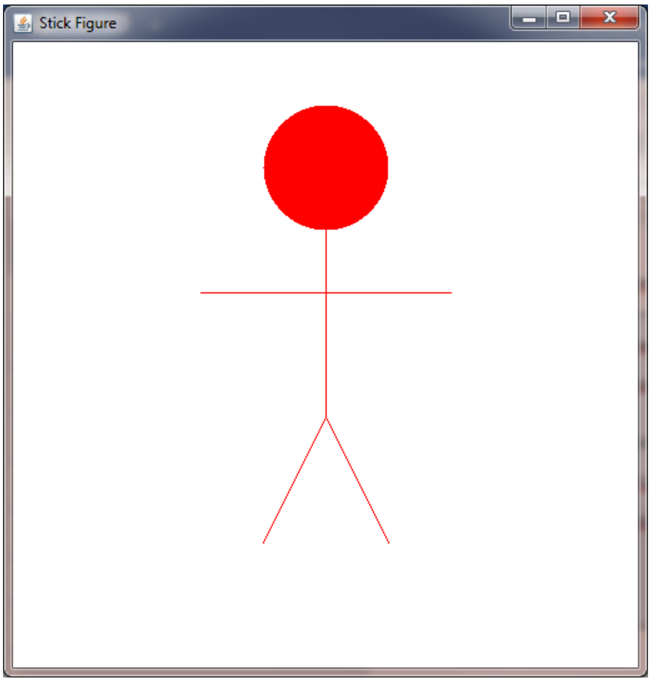
Your completed StickFigureExample file should like this:

```
package stickfigureexample;

public class StickFigureExample
{

    public static void main(String[] args)
    {
        Canvas stickFigureCanvas = new Canvas("Stick Figure", 500, 500);

        StickFigure stickFigure = new StickFigure(stickFigureCanvas);
        stickFigure.drawStickFigure();

    }

}
```

Press the play button (below) at the top of the screen to run the program

▷

The following should be displayed:



So, this is the first stage of the development completed. The next thing we need to do is to adapt both files to allow us to generate a random colouring for the stick figure.

In the StickFigure class, change the class so that it has an instance variable of type Color, which is passed as a parameter. You will then need to change the setForegroundColor method parameter to use the instance variable you have just created.

The completed, adapted class file should look like the below:

```
package stickfigureexample;

import java.awt.*;  // allows us to use the Color class to define the stick figure's colour

/**
* Class to define the drawing of a stick figure in the middle of the provided window
*/
public class StickFigure
{
    private Canvas window;
    private Color colour;

    /**
    * Constructor for StickFigure
    * @param win The window to draw the stick figure on
    * @param stickFigureColour The colour of the stick figure
    */
    public StickFigure(Canvas win, Color stickFigureColour)
    {
        window = win;
        colour = stickFigureColour;
    }

    /**
    * Method to draw the stick figure
    */
    public void drawStickFigure()
    {
        // set the foreground colour
        window.setForegroundColor(colour);

        // head
        window.fillCircle(250, 100, 100);

        // body
        window.drawLine(250, 150, 250, 300);

        // arms
        window.drawLine(150, 200, 350, 200);

        // left leg
        window.drawLine(250, 300, 200, 400);

        // right leg
        window.drawLine(250, 300, 300, 400);
    }

}
```

We want to minimise the amount of code which is included in the main method. Therefore, we will create a class which will generate the random colour which will be used to draw the stick figure. Later in this example, the class will be used to get the users input for the window size.

Create a new class called "PrepareStickFigureInfo". This class will have no instance variables, and will not require a constructor.

Now that the stick figure can be any colour, we need to create the random colour generator. Firstly, import the java.util and java.awt packages, so that we have access to the Random and Color classes. Then create a method called getRandomColour(), which as a return type of Color and doesn't take any parameters.

```
package stickfigureexample;

import java.awt.*;
import java.util.*;

/**
 * Class which will generate the random colour required for the stick figure,
 * and get the window size user input
 *
 * @author AncientC
 */
public class PrepareStickFigureInfo
{
    /**
     * Method to generate a random colour for the stick figure
     * @return a random colour (either red, blue, or green)
     */
    public Color getRandomColour()
    {
         // code to generate a random colour
    }
}
```

In the getRandomColour() method, we need to first create a random object, which will allow us to generate a random integer. As we have 3 possible options, we want to give the random number generation a limit of 3.

```
Random random = new Random();

int randInt = random.nextInt(3);
```

We can now use if statements to decide which colour the stick figure will be:

```
if (randInt == 0)
{
     stickFigureColour = Color.RED;
}
else if (randInt == 1)
{
     stickFigureColour = Color.BLUE;
}
else
{
     stickFigureColour = Color.GREEN;
}
```

You will notice that we used 0, 1, and else. This is because the `random.nextInt(3)` line will produce either 0, 1, or 2.

The completed PrepareStickFigureInfo class should be as follows:

```
package stickfigureexample;

import java.awt.*;
import java.util.*;

/**
 * Class which will generate the random colour required for the stick figure,
 * and get the window size user input
 * @author AncientC
 */
public class PrepareStickFigureInfo
{
    /**
     * Method to generate a random colour for the stick figure
     * @return a random colour (either red, blue, or green)
     */
    public Color getRandomColour()
    {
        Random random = new Random();

        int randInt = random.nextInt(3);

        if (randInt == 0)
        {
            return Color.RED;
        }
        else if (randInt == 1)
        {
            return Color.BLUE;
        }
        else
        {
            return Color.GREEN;
        }
    }
}
```

Now the main method needs to be adapted to give the colour of the stick figure to the object as a parameter. Create a new object of type PrepareStickFigureInfo called info. Create this object before you create the Canvas object.

```
PrepareStickFigureInfo info = new PrepareStickFigureInfo();
```

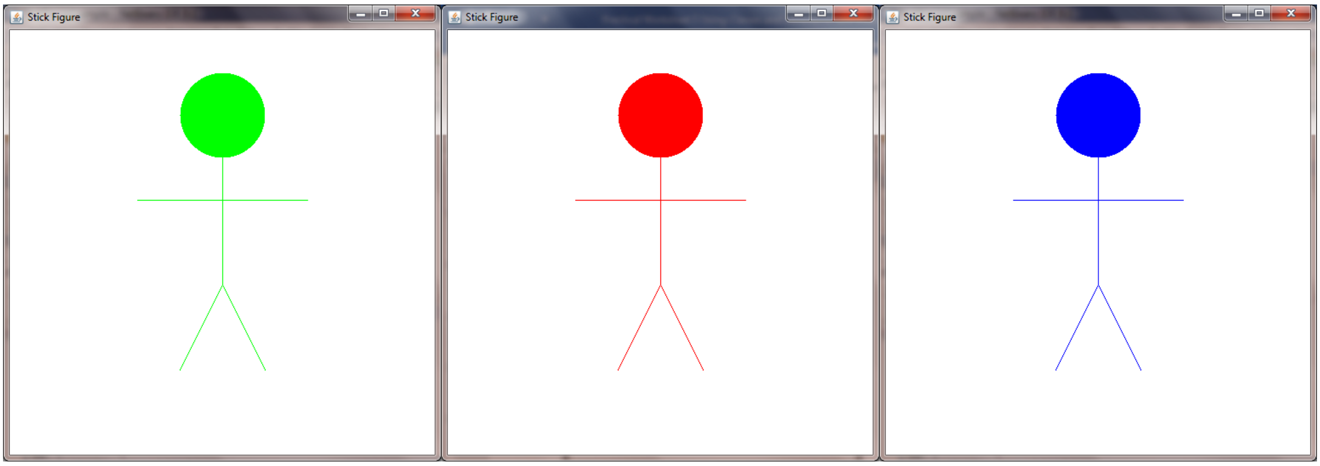
You will also need a new variable of type Color called stickFigureColour. You will need to import the java.awt package. The stickFigureColour will be assigned using the result of the getRandomColour method from the info object.

```
Color stickFigureColour = info.getRandomColour();

StickFigure stickFigure = new StickFigure(stickFigureCanvas, stickFigureColour);
```

The completed class file containing the main method should be as follows:

```
package stickfigureexample;

import java.awt.*; // so we can use colours

public class StickFigureExample {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args)
    {
        PrepareStickFigureInfo info = new PrepareStickFigureInfo();

        Canvas stickFigureCanvas = new Canvas("Stick Figure", 500, 500);

        Color stickFigureColour = info.getRandomColour();

        StickFigure stickFigure = new StickFigure(stickFigureCanvas, stickFigureColour);
        stickFigure.drawStickFigure();

    }

}
```

Run the main method. You should get a stick figure drawn in a pop up window. You might want to run it a few times, to see the different colours.



We have now completed the first 2 stages of the process: created the stick figure, and adapted it to be any colour we want. The next step is to make it scalable, so that any window size can be used.

Open the StickFigure class, and create a new instance variable called scale. It will need to be a double type, so that we can enter decimal values. This instance variable will also need to be passed as a parameter.

We then need to multiply all the values which are used to create the stick figure by the scale to make them smaller or larger (depending on the number passed as the parameter).

Your completed stick figure class should be as follows:

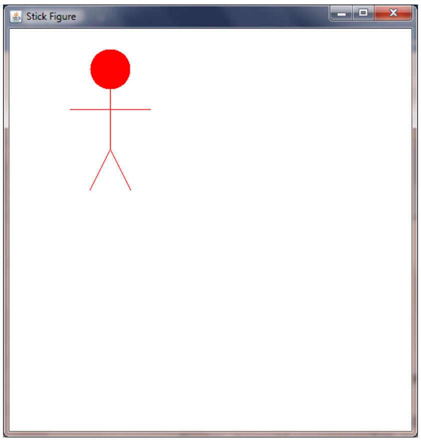
```
package stickfigureexample;

import java.awt.*;  // allows us to use the Color class to define the stick figure's colour

/**
* Class to define the drawing of a stick figure in the middle of the provided window
*/
public class StickFigure
{
    private Canvas window;
    private Color colour;
    private double scale;

    /**
    * Constructor for StickFigure
    * @param win The window to draw the stick figure on
    * @param stickFigureColour The colour of the stick figure
    */
    public StickFigure(Canvas win, Color stickFigureColour, double scaleFactor)
    {
        window = win;
        colour = stickFigureColour;
        scale = scaleFactor;
    }

    /**
    * Method to draw the stick figure
    */
    public void drawStickFigure()
    {
        // set the foreground colour
        window.setForegroundColor(colour);

        // head
        window.fillCircle(250*scale, 100*scale, 100*scale);

        // body
        window.drawLine(250*scale, 150*scale, 250*scale, 300*scale);

        // arms
        window.drawLine(150*scale, 200*scale, 350*scale, 200*scale);

        // left leg
        window.drawLine(250*scale, 300*scale, 200*scale, 400*scale);

        // right leg
        window.drawLine(250*scale, 300*scale, 300*scale, 400*scale);
    }

}
```

You will need to adapt the StickFigureExample class to use a scale when creating the stick figure object. Just to test that it is working, we will use 0.5 as our scale initially.

The completed StickFigureExample class should be:

```
package stickfigureexample;

import java.awt.*; // so we can use colours

public class StickFigureExample {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args)
    {
        PrepareStickFigureInfo info = new PrepareStickFigureInfo();

        Canvas stickFigureCanvas = new Canvas("Stick Figure", 500, 500);
        Color stickFigureColour = info.getRandomColour();

        StickFigure stickFigure = new StickFigure(stickFigureCanvas, stickFigureColour, 0.5);
        stickFigure.drawStickFigure();

    }
}
```

If you run the main method, you should have a window which shows a smaller Stick Figure.



The final step we need to carry out is to get the user input to size the window. We will also use that input to calculate the stick figure's scale so that it is always positioned in the middle of the window at an appropriate size.

In the PrepareStickFigureInfo class, add a method which will use the KeyboardInput class to get the window dimensions. We assume that the window is square, so only the height is required. The method will return an integer value (as the Canvas class requires integers as part of its constructor).

```
public int getWindowSize()
{
    KeyboardInput keyboardInput = new KeyboardInput();

    System.out.print("Please enter the window height: ");
    int windowSize = keyboardInput.getInputInteger();
    System.out.println("");
    System.out.println("The window will be " + windowSize + " x " + windowSize);

    return windowSize;
}
```

In the main method, add an extra variable before the Canvas object is created. This variable will be initialised using the value returned by the method you have just written:

```
int winSize = info.getWindowSize();
```

We can now create another variable, which is of type double, to calculate the scale for the stick figure. Initially, we calculated all the coordinates for the stick figure using a 500 x 500 window. Therefore, if we divide the entered window size by 500, we will get the appropriate scale for the stick figure. Make sure you divide by 500.0, otherwise integer division will be carried out, so you will get the wrong scale.

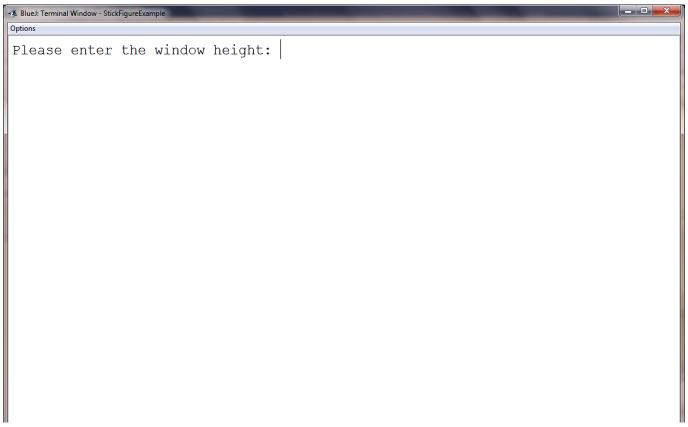
```
double scale = winSize / 500.0;
```

You will then need to adapt the rest of the main method so that the window size uses the winSize variable (in the creation of the Canvas object) and the scale in the StickFigure creation uses the new variable.

The completed StickFigureExample class should be:

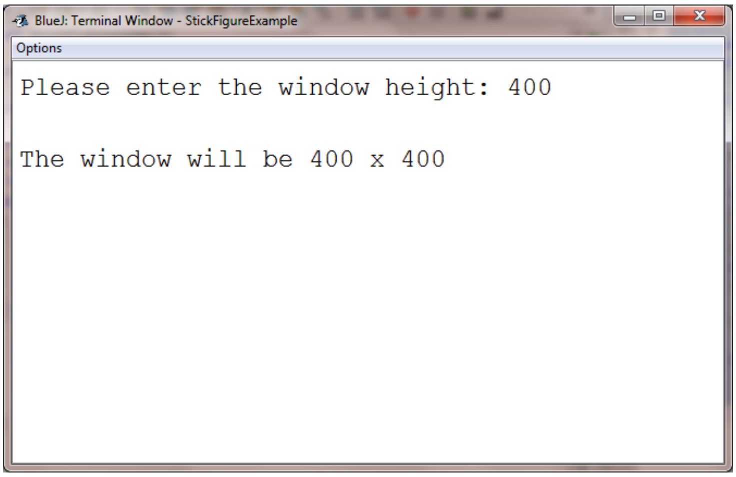
```
package stickfigureexample;

import java.awt.*; // so we can use colours

public class StickFigureExample {

  /**
   * @param args the command line arguments
   */
  public static void main(String[] args)
  {
     PrepareStickFigureInfo info = new PrepareStickFigureInfo();

     int winSize = info.getWindowSize();
     double scale = winSize / 500;

     Canvas stickFigureCanvas = new Canvas("Stick Figure", winSize, winSize);
     Color stickFigureColour = info.getRandomColour();

     StickFigure stickFigure = new StickFigure(stickFigureCanvas, stickFigureColour, scale);
     stickFigure.drawStickFigure();

  }
}
```

If we run the main method, we should get a terminal window popup, which requests the window height



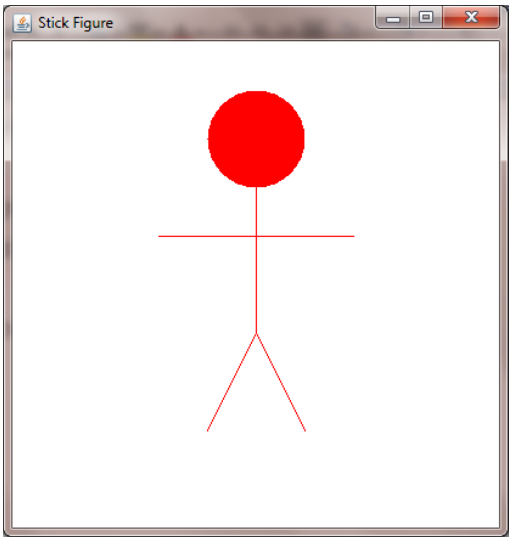
```
Please enter the window height: 
```

Enter a value for the height (in this example we have put 400 as the window size)



```
Please enter the window height: 400

The window will be 400 x 400
```

The stick figure window will then be displayed with the scaled stick figure:

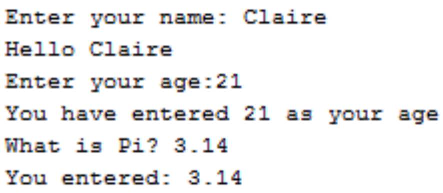


Run the main method a few times, with different window sizes, to see how it works.

3. Practice Questions

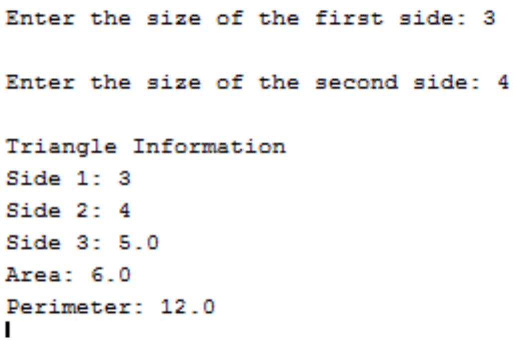
The following questions will allow you to practice using classes in Java, with a particular focus on the Canvas and KeyboardInput classes which are available on Moodle. Make sure that you don't put too much code within the main method – just like we have just done in the example.

1. Using the KeyboardInput class, write a program which asks the user for their name, then says hello to them using the name they have entered. Also experiment with getting the user to input integers and doubles. Each time the user enters some data, make sure that the input is repeated in the console.
   For example:

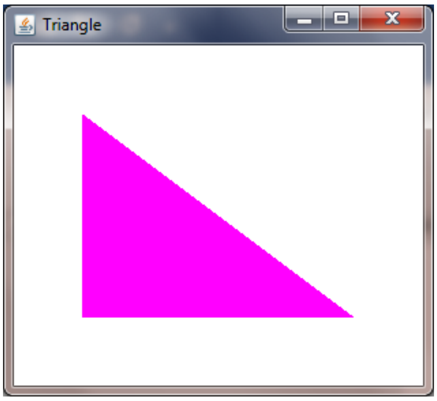


```
Enter your name: Claire
Hello Claire
Enter your age:21
You have entered 21 as your age
What is Pi? 3.14
You entered: 3.14
```

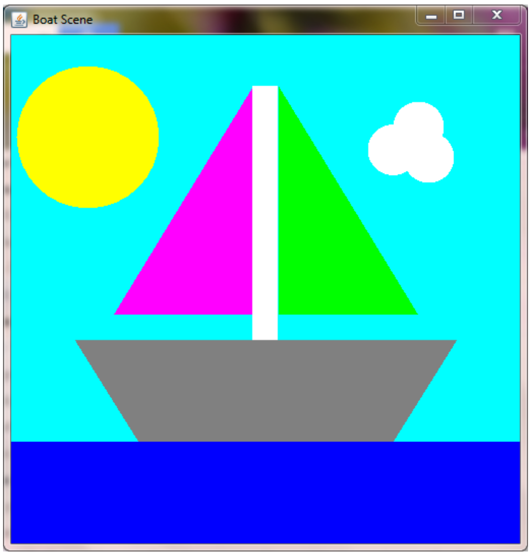
2. Create a program which will ask the user for 2 sides of a triangle to create a right angled triangle. You can assume they will always enter whole numbers. The program will then print the length of the third side, the perimeter, and area of the triangle to the console. It will also draw a triangle with the 3 sides in the correct proportion, in a colour of your choice.



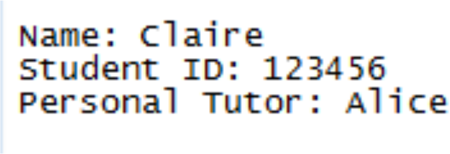
```
Enter the size of the first side: 3

Enter the size of the second side: 4

Triangle Information
Side 1: 3
Side 2: 4
Side 3: 5.0
Area: 6.0
Perimeter: 12.0
```



3. Draw the below boat scene, using the provided Canvas class. The window it uses should be 500 x 500, and called Boat Scene. I would recommend that you plan the design on squared paper before attempting to code it (just like we did during the lecture). You can choose whatever colour sails you want!

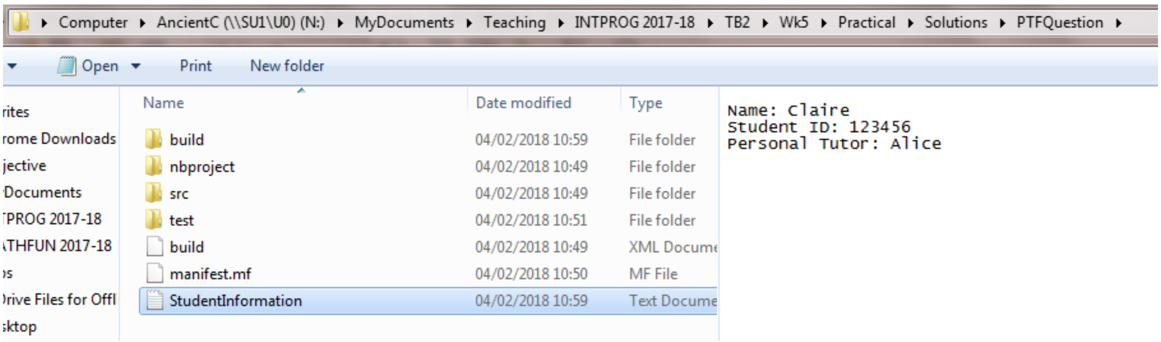


4. Using the createNewFile() method in the PrintToFile class on Moodle, create a file called "StudentInformation.txt". Using the correct method within the class, enter the following information into the file:
    - Your name
    - Your student ID number
    - Your personal tutor



```
Name: Claire
Student ID: 123456
Personal Tutor: Alice
```

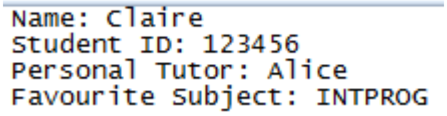
<u>Hint:</u> the "\n" notation will also work in Java.
<u>Hint 2:</u> You will need to navigate to the same location as you have stored your NetBeans package to find the file created.


Computer ▸ AncientC (\\SU1\U0) (N:) ▸ MyDocuments ▸ Teaching ▸ INTPROG 2017-18 ▸ TB2 ▸ Wk5 ▸ Practical ▸ Solutions ▸ PTFQuestion ▸
Open
Print
New folder
rites
rome Downloads
jective
Documents
PROG 2017-18
THFUN 2017-18
s
rive Files for Offl
sktop

Name	Date modified	Type
build	04/02/2018 10:59	File folder
nbproject	04/02/2018 10:49	File folder
src	04/02/2018 10:49	File folder
test	04/02/2018 10:51	File folder
build	04/02/2018 10:49	XML Docume
manifest.mf	04/02/2018 10:50	MF File
StudentInformation	04/02/2018 10:59	Text Docume

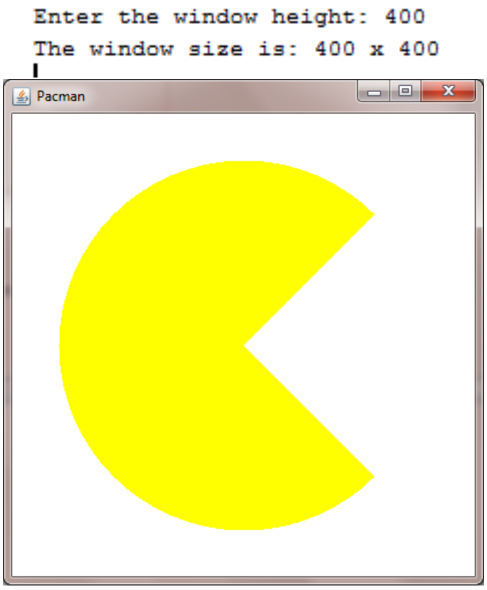
```
Name: Claire
Student ID: 123456
Personal Tutor: Alice
```

5. Using the addToFile() method in the PrintToFile class to add your favourite subject to the "StudentInformation.txt" file.



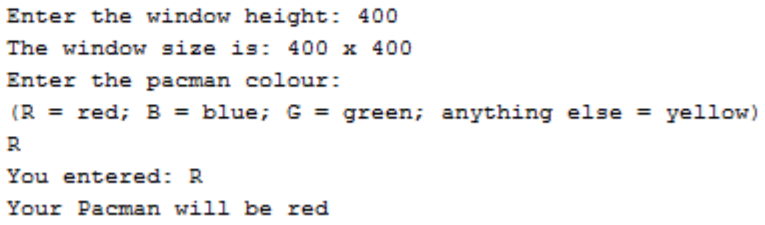
```
Name: Claire
Student ID: 123456
Personal Tutor: Alice
Favourite Subject: INTPROG
```

6. Create a program which asks the user for the height of a window, and draws a Pacman in the middle of the window. In a similar manner to the example, the Pacman needs to be scaled based on the inputs of the user so that it fills the window.



```
Enter the window height: 400
The window size is: 400 x 400
```

7. Adapt the Pacman program so that the user can also enter a letter to decide what colour it should be. The following rules should apply:
    - R to make the Pacman red
    - B to make the Pacman blue
    - G to make the Pacman green
    - If the user enters an invalid letter, the Pacman should be drawn in the traditional yellow.



```
Enter the window height: 400
The window size is: 400 x 400
Enter the pacman colour:
(R = red; B = blue; G = green; anything else = yellow)
R
You entered: R
Your Pacman will be red
```