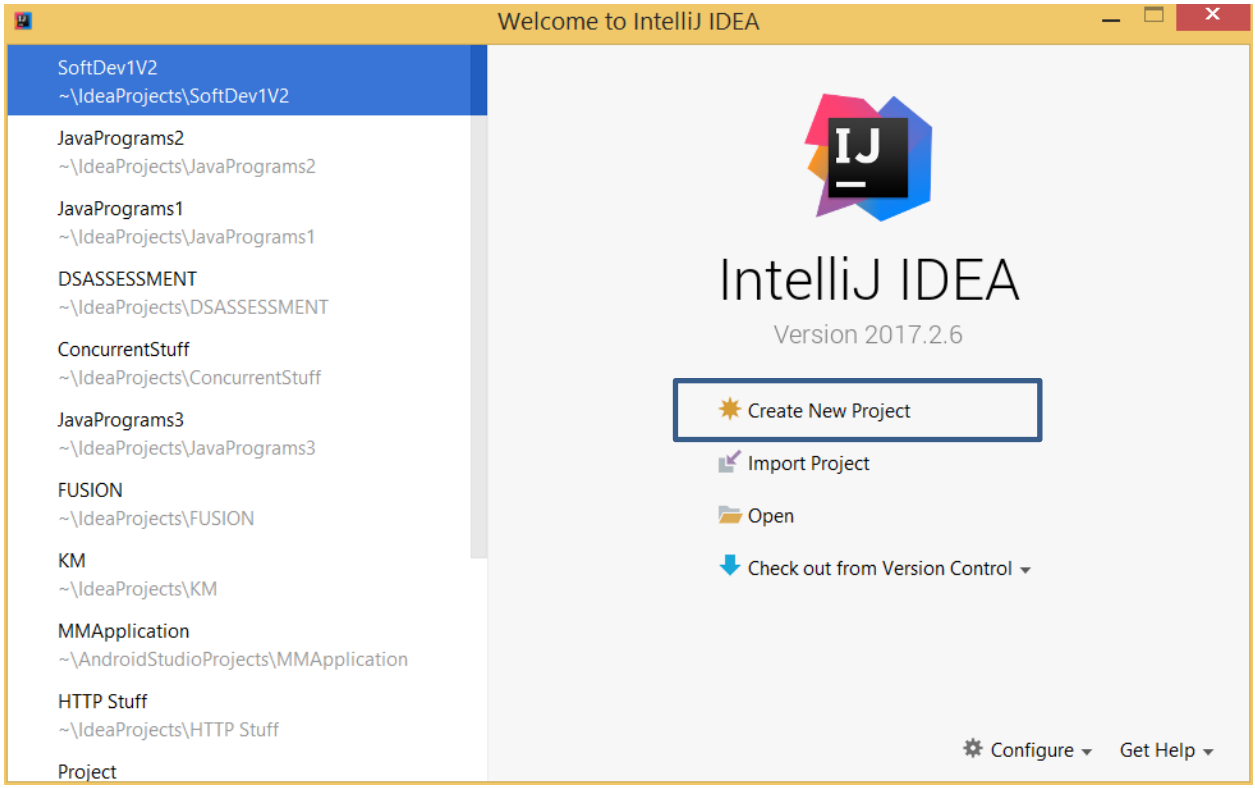
SOFTWARE DEVELOPMENT I

WEEK 1 LAB 1

1. CREATING A NEW PROJECT

When you run **IntelliJ**, you will see a screen similar to the following:



You will see that there is an option to create a NEW PROJECT. Projects in IntelliJ are used to bundle together related work. Hence, I would suggest that you should create a new PROJECT for each MODULE that you study.

There are 3 programming modules as follows, so I would suggest the following:

MODULE	**SUGGEST PROJECT NAME**
Software Development I	**SoftDev1**
Software Development II	**SoftDev2**
DataStructures	**DataStructures**

If you have followed the instructions for installing IntelliJ (previously circulated) you should already have created a Project so, if you wish you can create a NEW PROJECT with the title above (SoftDev1).

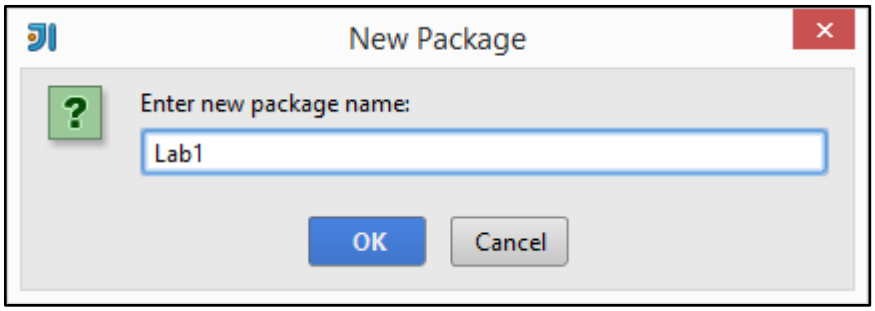
2. PACKAGES

Once you have a PROJECT created, you can create PACKAGES to reside within the PROJECT.

I would be inclined to create a new PACKAGE for each week (or each lab session). This allows you group your work and will make it easier if you want to locate a program that you have previously written.

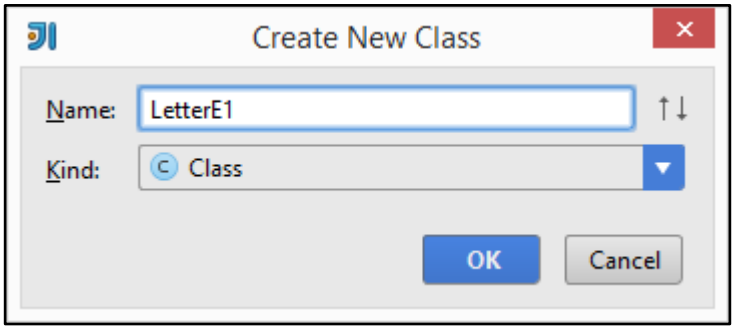
You can create a New Packageas follows:

- Double-click on **JavaPrograms** in the left window.
- Right click on the **src** folder.
- Select **New**
- Select **Package**
- Enter the name as **Lab1**
- Click on **OK**



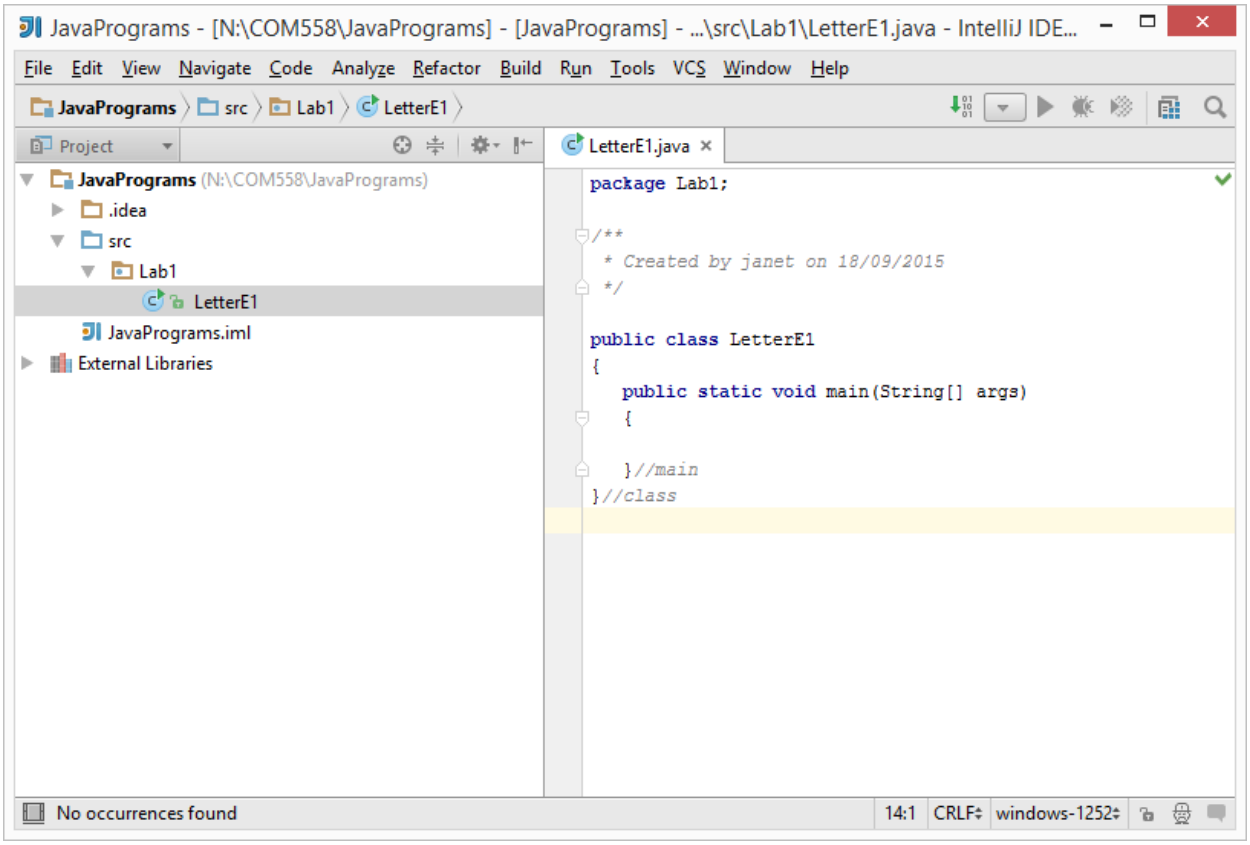
3. CREATING A PROGRAM (CLASS)

- Right click on the **Lab1** folder.
- Select **New**
- Select **Java Class**
- Enter Name as **LetterE1**
- Click on **OK**



The file is stored as **LetterE1.java**.

You should now see the something similar to the screen below:



4. DEFAULT SETTINGS

COMMENTS

- Let us examine the default settings:
  - Click on **File**, **Settings…**
  - Click on **Editor**
  - Click on **File and Code Templates**
  - Click on the **Includes** tab.
  - Click on **File Header**.
  - You should see information similar to that shown below:

```
/**

 *  Created by ${USER}on ${DATE}

*/
```

This comment will appear at the top of all programs that are created in IntelliJ.

${USER} will be replaced by the name of the logged in user.

${DATE} will be replaced by the current date.

CLASS STYLE

- Click on the **Templates** tab
- Click on **Class**. You should see information similar to that shown below:

```
public class ${NAME} {
   public static void main (String [] args) {

   }//main
}//class
```

This is the default structure of all classes created in IntelliJ.

${NAME} will be replaced by the Class name you have entered from the keyboard.

- Click on the **Cancel** button

THE LetterE1 PROGRAM

Amend the Letter E1 program already there by inserting the following statements under the main method header as shown below:

```
public static void main (String [] args) {

        System.out.println("******");
        System.out.println("*");
        System.out.println("*");
        System.out.println("*****");
        System.out.println("*");
        System.out.println("*");
        System.out.println("******");

}//main
```

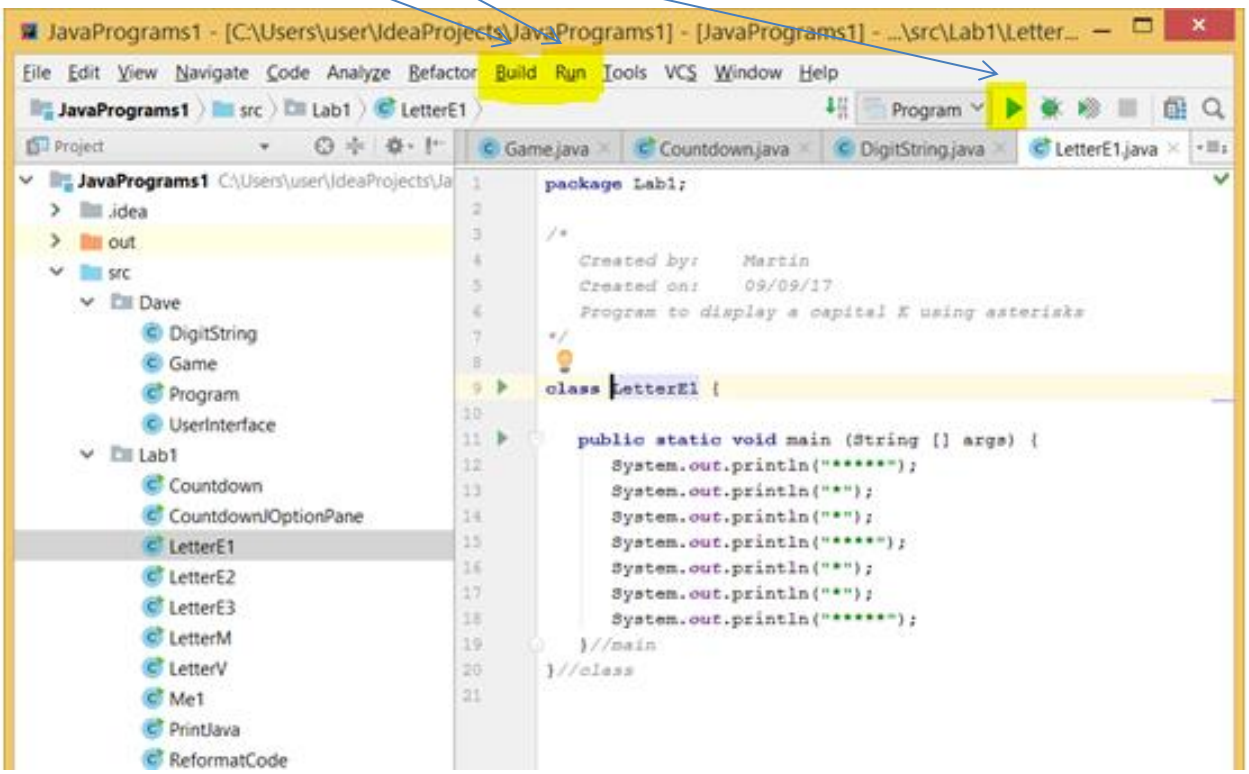
- Edit the File Header so that it includes a comment indicating what the program does. It should look similar to the following:

```
/**
 * Created by Martin on 09/03/2017
 * Simple program to print out the letter E using asterisks
 */
```

You have now completed your first program called LetterE1.

5. RUNNING YOUR PROGRAM

- Select **Build**, then **Run** from top menu bar. Click on **Run…** then select **LetterE1**



Your program should print out the letter E.

Once you have run the program once, you can click the **Run** button, or use **Shift + F10** as LetterE1 is the current program.

6. DEBUGGING A PROGRAM

Now that you have completed and run a java program, it is essential that you can develop your debugging skills.

What do we mean by debugging skills?

> "**Debugging** is a methodical process of finding and reducing the number of bugs, or errors/defects, in a computer program".

- Look at the first instruction in your program:

```
System.out.println("******");
```

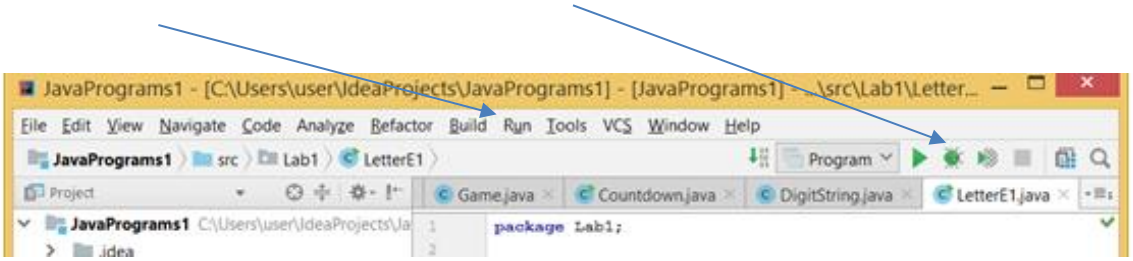
- Remove the **n** from the end of the **println** command. What colour does the command change to?

ANSWER:

- Why do think this is?

ANSWER:

- Select **Run** from top menu bar and then **Debug…** or click the **Debug** button or **Shift + F9**.



- What is the error message?

ANSWER:

- What does this tell you?

ANSWER:

- Look at the first instruction again in your program:

```
System.out.printl("******");
```

- Remove the **;** from the end of the line. Debug your program again. What is the error message?

ANSWER:

- What does it tell you?

ANSWER:

- To finish off correct the two errors you created in your program and save the file.

7. LOCATING YOUR JAVA FILES

- Open up a File Browser for example: My Computer

- Locate your **COM558** folder in your home drive (eg N:\COM558). You should see a **JavaPrograms** folder.

- Double click on the **JavaPrograms** folder. You should see a **src** folder.

- Double click on the **src** folder. This is where IntelliJ will store all your Java files. You should see a **Lab1** folder which will store all the programs created as part of Lab1. You will create a new package at each lab class so that you can easily find your programs.

- Double click on the **Lab1** folder. You should see the program you have just created (LetterE1.java). All java programs have the .java extension.

- Can you locate the class files? At present you will only have one class file – LetterE1.class. What is the path of the directory which contains the class files?

8. REFORMATTING JAVA CODE

- Create a program called **ReformatCode**. (right click on Lab1 and select **New**, **Java Class**. Type **ReformatCode** as the **Name:** Do not add **.java**).

- Insert the following statements under the main method header (start at the left margin and just keep typing – do not use the **Enter** key):

  System.out.println("I am testing the reformatting of code"); System.out.println("I am typing this paragraph without any structure"); System.out.println("After I have typed these statements into IntelliJ, I am going to use the Reformat command"); System.out.println("The reformatted code should make the program more readable");

- Run the program

- Click on **Code**, **Reformat Code…**

  The program should be reformatted to that of the default layout.

- Run your program again.

  The output should be the same as before.

9. ANOTHER PROGRAM

LetterE2.java

- Create the **LetterE2. java** program which was discussed in lectures.
- Run the program.
- Move the position of **\n** in some of the print() statements. Run the program again and make sure you understand the output generated.

EXERCISES

1. Create and run the following java programs which were discussed in lectures:
   - **LetterE3.java**
   - **Countdown.java**

   Make sure you understand the output generated.

2. Add spaces (inside and outside the double quotes) to **Countdown.java** and re-run the program.

3. Create a java program called **LetterV.java** which uses print() and/or println() statements and \t and/or \n to create the following output:

```
V                 V
   V           V
      V     V
         V
```

4. Create a java program called **LetterM.java** which uses print() and/or println() statements and \t and/or \n to create the following output:

```
M                 M
M  M           M  M
M     M     M     M
M        M        M
```

5. Create a java program called **PrintJava.java** which uses print() and/or println() statements and \t and/or \n to create the following output:

```
    J          A          V    V          A
    J         A A         V  V         A  A
J   J        AA AA        V V         AA  AA
 J  J        A    A        V          A    A
```

6. Create a program called **Me1.java** which uses print() and/or println() statements and escape characters to create the following output (substitute the details in italics with your own details):

Name:	*Martin McKinney*	Address:	*Room L142* *Ulster University*
Phone:	*Ext. 24495*		*Cromore Road* *Coleraine*
email:	*met.mckinney@gmail.com*		*BT52 1SA*
Height:	*5'9"*	Weight:	*82*kgs

7. Rewrite the Countdown program to display the output using **JOptionPane**.

   Save the program as **CountdownJOptionPane.java**.

8. Uploading Programs to **Blackboard**

   Upload **Me1.java** to Blackboard.

   The upload button is available in:

   Java, **Lab Exercises**, **Lab 1 – Introduction to Java**, **Lab 1 Upload**