Java Labs

Each weeks labs should be held in a separate folder: E.g. lab1.

Create this new folder each week in your user directory on the u drive - E.g. u:\software_development_1\lab1.

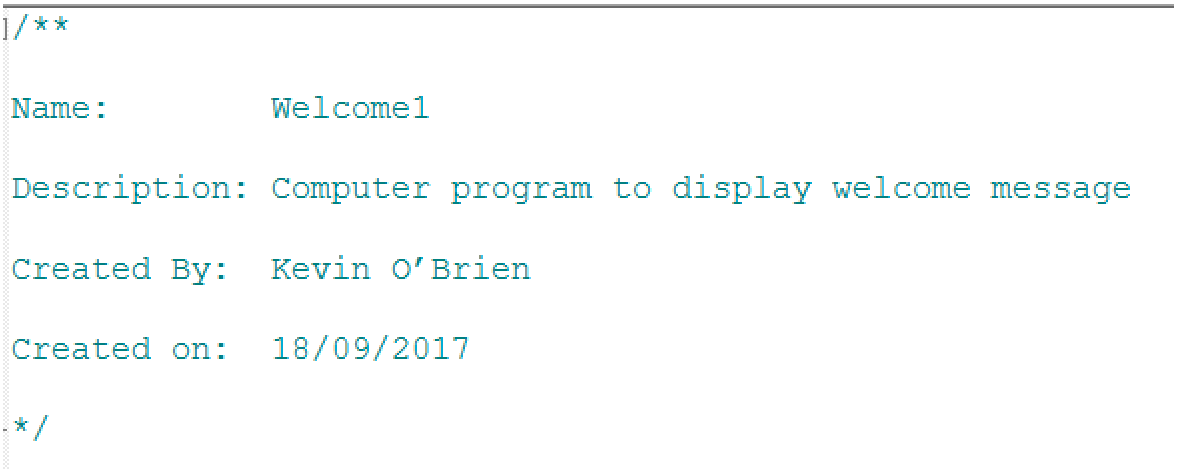
Save all files you work on in this lab to this folder.

All Exercises should be completed using TextPad.

To compile your code in TextPad, hit the ctrl key and 1 key. Hold down the ctrl key, while holding that key down press 1.

To run your compiled program, .class file, in TextPad, hit the ctrl key and 2 key. Hold down the ctrl key, while holding that key down press 2.

For all programs please insert a header similar to the following:



```
/**

Name:         Welcome1

Description:  Computer program to display welcome message

Created By:   Kevin O'Brien

Created on:   18/09/2017

*/
```

For all exercises, follow the instructions to implement the solution.

Introduction

In this weeks lab we will be focussing on creating basic computer programs to implement:

- Continue Statements
- Switch Statements
- Iterations
- Methods

Note that at the end of the end of the lab one or more of the exercises need to be submitted to your GitHub repository you created on week 19. Exercise(s) to submit to Git will be indicated to you by your lecturer.

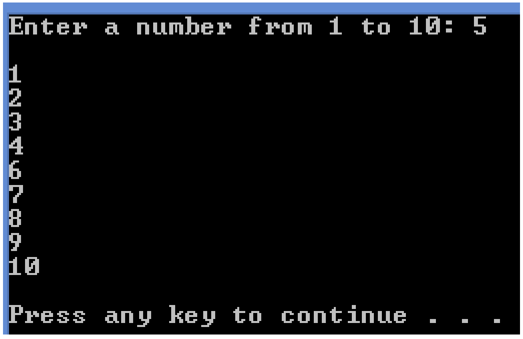
Section A

Exercises 1 to 6. Complete All.

Exercise 1: - Continue (Continue.java)

Write a program, Continue.java, which prompts a user to select a number from 1 to 10, inclusive. This number is then used to determine which iteration of a loop should not be completed. On each iteration of the loop, print out the value of the counter, except on the iteration which matches the number input by the user. The loop should iterate a maximum of 10 times.

Example output as follows:

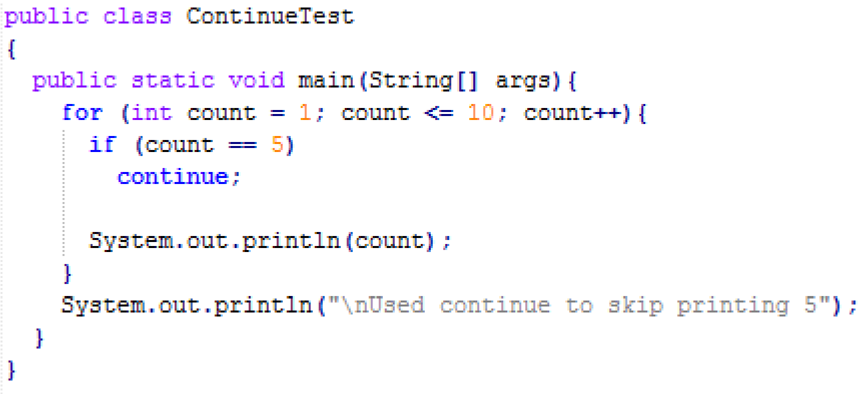


```
Enter a number from 1 to 10: 5

1
2
3
4
6
7
8
9
10

Press any key to continue . . .
```

Hint: Use the continue keyword.

Exercise 2 – Replace Continue (ContinueTest.java)

Rewrite the following program:



```
public class ContinueTest
{
  public static void main(String[] args){
    for (int count = 1; count <= 10; count++){
      if (count == 5)
        continue;

      System.out.println(count);
    }
    System.out.println("\nUsed continue to skip printing 5");
  }
}
```

So that instead of using continue statement to achieve the desired output, replace it with a structured equivalent.

Exercise 3 - Calculating Sales (Sales.java)

An online retailer sells five products whose retail prices are as follows:

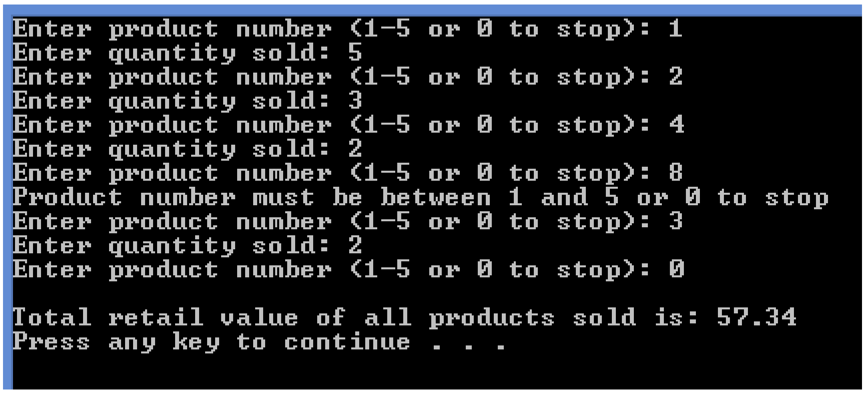
- Product 1, 2.98
- Product 2, 4.50
- Product 3, 9.98
- Product 4, 4.49
- Product 5, 6.87

Write a program, Sales.java, that reads a series of pairs of numbers as follows:

a. Product number
b. Quantity sold

Your program should use a switch statement to determine the retail price for each product. It should calculate and display the total retail value of all products sold. Use a sentinel-controlled loop to determine when the program should stop looping and display the final results.

An example output is as follows:



```
Enter product number (1-5 or 0 to stop): 1
Enter quantity sold: 5
Enter product number (1-5 or 0 to stop): 2
Enter quantity sold: 3
Enter product number (1-5 or 0 to stop): 4
Enter quantity sold: 2
Enter product number (1-5 or 0 to stop): 8
Product number must be between 1 and 5 or 0 to stop
Enter product number (1-5 or 0 to stop): 3
Enter quantity sold: 2
Enter product number (1-5 or 0 to stop): 0

Total retail value of all products sold is: 57.34
Press any key to continue . . .
```

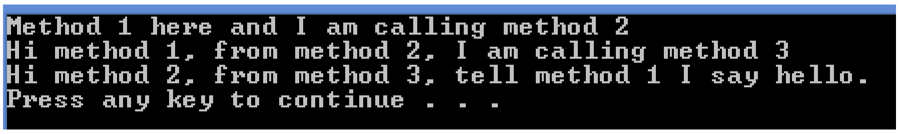
Exercise 4 – Methods (Methods.java)

Write a program, Methods.java, which includes the following methods:

```
public void method1(){
  System.out.println("Method 1 here and I am calling method 2");
} //end method1

public void method2(){
  System.out.println("Hi method 1, from method 2, I am calling method 3");
} //end method2

public void method3(){
  System.out.println("Hi method 2, from method 3, tell method 1 I say hello.");
} //end method3
```

Modify these methods so that method1 calls method2 and that method2 calls method3, so that you see the outputs from System.out.println in each method.

You will need a main method to create an instance of Method and invoke method1 using that instance.
The output should be as follows:

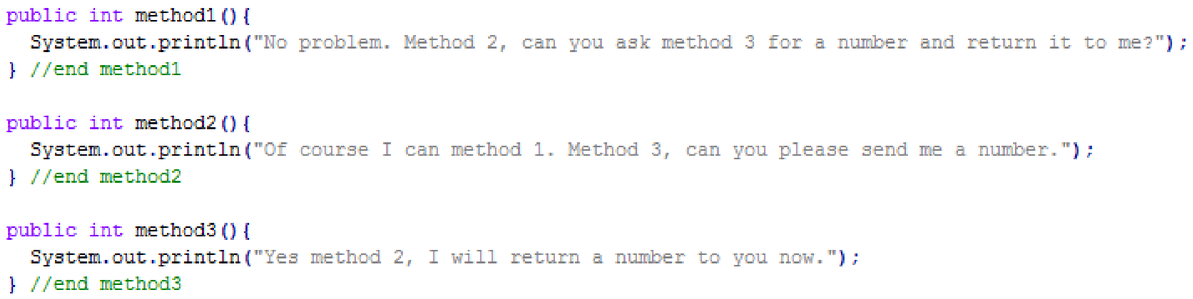


```
Method 1 here and I am calling method 2
Hi method 1, from method 2, I am calling method 3
Hi method 2, from method 3, tell method 1 I say hello.
Press any key to continue . . .
```

Exercise 5 – Return values from methods (ReturnMethods.java)

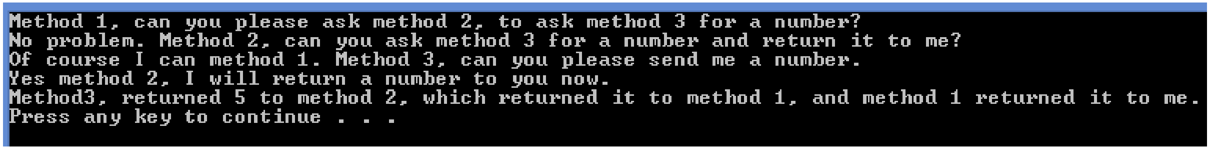
Write a program, ReturnMethods.java, which is similar to exercise 4, except that the methods return a value. The value should be of type int and return the number 5. The number 5 should be specified in method3 only, and returned to method2 which returns it to method1.

Some code below to help you get started:



```
public int method1(){
  System.out.println("No problem. Method 2, can you ask method 3 for a number and return it to me?");
} //end method1

public int method2(){
  System.out.println("Of course I can method 1. Method 3, can you please send me a number.");
} //end method2

public int method3(){
  System.out.println("Yes method 2, I will return a number to you now.");
} //end method3
```

An example output of this program is as follows:



```
Method 1, can you please ask method 2, to ask method 3 for a number?
No problem. Method 2, can you ask method 3 for a number and return it to me?
Of course I can method 1. Method 3, can you please send me a number.
Yes method 2, I will return a number to you now.
Method3, returned 5 to method 2, which returned it to method 1, and method 1 returned it to me.
Press any key to continue . . .
```

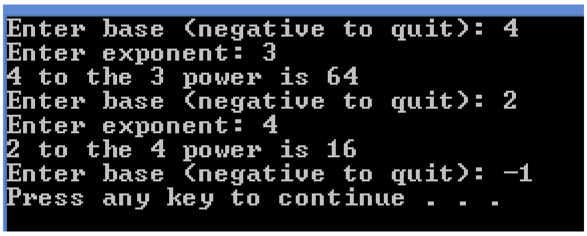
Exercise 6 – Power of Method (Power.java)

Write a program, Power.java, which includes a method, integerPower(base, exponent) that returns the value of:

$\text{base}^{\text{exponent}}$

For example, integerPower (3, 4) calculates 3^4 (or 3 * 3 * 3 * 3). Assume that exponent is a positive, nonzero integer and that base is an integer. Use a for or while statement to control the calculation. Do not use any Math class methods. Incorporate this method into an application that reads integer values for base and exponent and performs the calculation with the integerPower method.

An example output is as follows:



```
Enter base (negative to quit): 4
Enter exponent: 3
4 to the 3 power is 64
Enter base (negative to quit): 2
Enter exponent: 4
2 to the 4 power is 16
Enter base (negative to quit): -1
Press any key to continue . . .
```