

## Lab 5 – Repetition

Program Design Sheets should be completed for problems 3, 4, 5, 6, 8, 9, 12 and 16.

Create a new package called **Lab5** to store all the java programs you create in this lab class.

### FOR loops

1. Type in and run **AddNumbers1.java** and **OddNumbers.java** discussed in lectures.
  - 1.1 Amend AddNumbers1 to read in the number of values which are to be added up (rather than having a constant HOWMANY).
2. Write a Java application called **TestForLoop.java** which prints out the numbers 1 to 10, each on a separate line.
  - 2.1 Change the components of the **for** statement in your previous program so that each of the following will be printed (re-run the program each time):
    - Numbers 15 to 25
    - Numbers 10 to 1
    - Multiples of 2 from 2 to 20
    - Multiples of 3 from 30 to 12
    - Multiples of 5 from 35 to 75
3. Write a Java application called **KilosToPounds.java** which prints out the following table of kilograms from 5 to 100 (in increments of 5) and their equivalent pounds.

<b>Kilograms</b>	<b>Pounds</b>
5	11.0
10	22.0
15	33.0
...	
95	209.0
100	220.0
4. Write a Java application called **Average.java** which asks the user how many numbers they are going to enter, then reads in the numbers and calculates and outputs the average.

Test your program with the 6 values 64, 121, 400, 99, 0, 45.

## Lab 5 – Repetition

5. Write a Java application called **Divisor.java** which prompts for a **start** number, a **finish** number (assume  $\text{start} < \text{finish}$ ) and a **divisor**, then prints out all the numbers between **start** and **finish** (inclusive) which are evenly divisible by the divisor. eg

start = 25, finish = 36, divisor = 3

Output: 27

30

33

36

- 5.1 Extend the program to output the total number of numbers evenly divisible by the divisor and the average of these numbers. The output for the above data would be:

Total number of multiples of 3 is 4

Average of these 4 numbers is 31.50

Test your program with the following data:

Start = 25      Finish = 36      Divisor = 4

Start = 3      Finish = 24      Divisor = 5

Start = 100      Finish = 200      Divisor = 20

6. Write a Java application called **FizzBuzz2.java**. This program reads in 2 integers from the keyboard representing a **start** number and a finish number (assume  $\text{start} < \text{finish}$ ) and outputs FIZZ, BUZZ, FIZZ BUZZ or the number for all numbers from **start** to **finish** (inclusive) according to the following rules:

Number is evenly divisible by 3      **FIZZ**

Number is evenly divisible by 7      **BUZZ**

Number is evenly divisible by 3 and 7      **FIZZ BUZZ**

In all other cases      **The number itself**

The output for **start** =10, **finish** = 25 would be:

10	11	<b>FIZZ</b>	13	<b>BUZZ</b>	<b>FIZZ</b>	16	17
<b>FIZZ</b>	19	20	<b>FIZZ BUZZ</b>	22	23	<b>FIZZ</b>	25

Test your program with the following data:

Start = 1      Finish = 20

Start = 52      Finish = 70

Start = 100      Finish = 120

Start = 150      Finish = 199

## DO ... WHILE loops

7. Type in and run **Dice.java** and **DisplayMenu.java** discussed in lectures.
8. Write a Java application called **FindNumber.java** which randomly guesses a number between 0 and 9 (inclusive) until it guesses a predefined number (stored as a constant in the program). Output the number of attempts the computer makes at guessing the correct number.

Test your program several times.

9. A charity is trying to raise £500 for a project. Write a Java application called **Donations.java** which accepts variable donations from an unknown number of donors. As soon as the £500 target is reached the charity wants to announce that they have reached their target by highlighting the following information:

How many donations were received

Who donated the largest amount

Value of the largest donation

Test your program with the appropriate data.

## WHILE loops

10. Type in and run **Factors.java** and **MultiplesOfThree.java** which were discussed in lectures.
11. Create a Java application called **AddNumbers4.java**. Type in the code given for **AddNumbers3.java**. Run the program and make sure you understand how it works. Extend your program to count how many numbers have been entered and then work out and output the average.

Test your program with the following data:

2 4 6 2 8 7 -999

87 34 -8 56 -7 0 4 -66 23 -999

12. Write a Java application called **LargestNumber.java**. This program should read in a list of numbers terminated by -999 and find and output the largest number in the list.

Test your program with the same data as the previous exercise.

## Nested Loops

13. Type in and run **NestedForX.java** and **Multiplication.java** discussed in lectures.

14. Write a Java application called **NestedForLoops1.java**. This program should output the following (use nested for loops):

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

15. Write a Java application called **NestedForLoops2.java**. This program should output the following (use nested for loops):

```

                1
            2 1 2
        3 2 1 2 3
    4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```

16. Write a Java application called **NumberMenu1.java**. which will prompt the user to enter a number then display a menu:

1. Test if zero
2. Find square root
3. Test if Odd or Even
4. Enter Another Number or Exit

Depending on the option chosen, the program will carry out the appropriate action and print appropriate output. If option 4 is chosen the system will prompt the user if they want to enter another number or exit the program.

Test your program.