# 4CS001 Workshop 5

## Exercise 1 – while loop

Copy the following code into your editor and then compile and run the program.

**public class Week05Ex1 {**

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

**System.out.println("Start of program");**

**int counter = 0;**

**while(counter < 4) {**

**counter=counter+1;**

**System.out.println("Counter = "+counter);**

**}**

**System.out.println("End of Program");**

**System.out.println("Counter = "+counter);**

**}**

**}**

Choose a different finish value for the counter, compile and run the program again to check that it works.

## For loop

27

Another alternative iterative construct to while loops.

**for(int i = 0; i < 5; i=i+1) {**

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

**}**

initialisation statement

loop condition

Has the same effect as:

loop control variable is **i**

Often written as **i++**

increment statement

**int i = 0;**

**while (i<5){**

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

**i=i+1;**

**}**

This is TASK 6 of your WORKBOOK. Ensure it is marked by the end of week 6.

## Exercise 2 – for loop

Amend your program Week05Ex1 to use a for loop instead of a while loop, and rename your program Week05Ex2. Compile and run the program to check that it works.

This is TASK 6 of your WORKBOOK. Ensure it is marked by the end of week 6.

## Exercise 3 - Vending Machine Change Output

Based on your earlier experience of calculating how many of each coin a vending machine should dispense, write a program that prints a row for each value of change between 0 and 99 and columns for the amount of change required. For example the start of the table should look like the following.

**Change 50p 20p 10p 5p 2p 1p**

**0 0 0 0 0 0 0**

**1 0 0 0 0 0 1**

**2 0 0 0 0 1 0**

**3 0 0 0 0 1 1**

**4 0 0 0 0 2 0**

**5 0 0 0 1 0 0**

This is TASK 7 of your WORKBOOK. Ensure it is marked by the end of week 8.

## Exercise 4 - Credit Card Problem 1

* You have a credit card bill for £500, but can only afford to pay £100 pounds per month.
* Each month the bank will charge 10% on any outstanding balance.
* How much do you still owe after 3 months payments?
* Design an activity diagram to represent an algorithm for answering these questions.
  + Convert the activity diagram to code and implement it.
  + Use Jeliot to help perform a dry run on your code.
* HINT – A common trick in programming is to solve a new problem by reusing a solution to a similar problem.
  + How is the credit card exercise like the bouncing ball problem?
  + How is it different?

This is TASK 7 of your WORKBOOK. Ensure it is marked by the end of week 8.

## Exercise 5 - Credit Card Problem 2

* You have a credit card bill for £500, but can only afford to pay £100 pounds per month.
* Each month the bank will charge 10% on any outstanding balance.
* How many months will it take you to pay off the debt?
* How much will you actually have to pay?
* By amending your solution to the previous exercise:-
  + Draw a new activity diagram
  + Convert the activity diagram to code and implement it.
  + Use Jeliot to help perform a dry run on your code.

## Exercise 6 - Conversion Tables

Write a program to ask the user for a starting temperature between 0 and 10 Celsius and a finish temperature between 20 and 30 Celsius, then print out a temperature conversion table using System.out.println to produce output like :-

Degrees Celsius Degrees Fahrenheit

0 32

1 33.8

…… …..

30 86

You can use the "\t" escape sequence to print in neat columns.

You can use the formula covered in lecture week 3 for calculating Celsius from Fahrenheit, or details of the calculations required for conversion may be found on the Internet, e.g. <http://www.csgnetwork.com/tempconvjava.html>

## Exercise 7 - Number guessing game (while loop and selection exercise)

User tries to guess a number between 1 and 10 that the computer has thought of. Each time the User guesses the computer should answer “Higher”, “Lower” or “You got it”. This should continue until the User gets the number right, at which point the computer should tell the User how many guesses they took.

For example:-

Computer thinks of 2.

The User guesses 3 Computer answers **Lower**

The User guesses 1 Computer answers **Higher**

The User guesses 2 Computer answers **You got it**

**You got my number in 3 guesses**.

To create the computer’s number use the instruction

int myNumber = (int) (10\*Math.random());

This is a more complicated exercise. You will probably need to design a solution in an Activity Diagram first. It can be done using a while loop and if then else test inside the while loop.

## Exercise 8 - Patterns Exercises

1. This exercise uses nested for loops to print a rectangle of \* characters on the console. Examine the code below.

**/\*\***

**\* Patterns Exercise 1.**

**\***

**\* @author Gordon Branson**

**\*/**

**public class Exercise1**

**{**

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

**int numRows = 5;**

**int numCols = 6;**

**for (int row = 1; row <= numRows; row++){**

**for (int col = 1; col <= numCols; col++){**

**System.out.print("\*");**

**}**

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

**}**

**}**

**}**

Do you understand the code?  
Modify the program so that it requests the number of rows and columns from the user. You should use JOptionPane for this. Compile, Run and Test your modified program.

1. This exercise uses nested for loops to print a triangle of \* characters on the console. Examine the code below.

**/\*\***

**\* Patterns Exercise 2.**

**\***

**\* @author Gordon Branson**

**\*/**

**public class Patterns2 {**

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

**int numRows = 5;**

**for (int row = 1; row <= numRows; row++){**

**for (int col = 1; col <= row; col++){**

**System.out.print("\*");**

**}**

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

**}**

**}**

## }

Do you understand the code?  
Modify the program so that it requests the width (number of columns) from the user. You should use JOptionPane for this. Compile, Run and Test your modified program.

3) Using the previous exercises as a guide, write a new program (Class) to draw an upside down triangle like that shown below.

**\*\*\*\*\*\***

**\*\*\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

4) Using Exercise 3 as an example, write a new program to draw the following shape.  
  
(Assume the user has entered the value 4 for the width)

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

5) Using Exercise 4 as an example, write a new program to draw the following shape.  
  
(Assume the user has entered the value 4 for the width)

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\***

**\*\***

**\***

6) Using Exercise 5 as an example, write a new program to draw the following shape.  
  
(Assume the user has entered the value 7 for the width)

**\***

**\*\*\***

**\*\*\*\*\***

**\*\*\*\*\*\*\***

**\*\*\*\*\***

**\*\*\***

**\***

Note – This exercise will only work properly if the user enters an odd number for the width.

Can you improve on the program to ensure that the user *only* enters an odd number (i.e. validate the input so that it repeatedly asks the user to enter a number while the number entered is even)?

Can you go further and make sure that the user only enters numbers in a range of, say, 3 to 15?