# **Fundamentals: Recap - Exercises**

## **Objective**

This is a coding practice that requires using only skills learnt in block 1. The main aim of this exercise is to remind you of the basic knowledge you learned in the previous course.

## **Step by Step**

1. Create a new project
2. Create a class called QA with a main() method

## Calculate the Area and perimeter of a rectangle:

1. Create a method called getArea() and another called getPerimeter() in the QA class.
2. Each method should have two integer parameters for width and height.
3. Ask the user to input the width and height of a rectangle in the main() method.
4. Call the methods in step 3 to get the area and perimeter and then display these values using a formatted print (printf method)

## Create marks for a an array of grades

1. Create and integer array of grades containing the following numbers: 90,30,13,67,85,87,50,45,51,72,64,69,59,17,22,23,44,25,16,67,85,87,50,45,51,72,59,14,50,55
2. create a method to print the grade and the award using the following rules:

Less than 50 Fail

between  50..60 (inclusive) Pass

between  61..70 (inclusive) Merit

between  71..100 (inclusive) Distinction

## Create a class structure for QA training

1. Create a class called **TrainingCourse**
2. Add the following fields

string **name**, int **duration**, Date **startDate**;

**Tip:** You'll need to import *java.util.Date* for objects of type Date

1. Create a constructor to pass values in for the a course's name and duration.

Set the startDate field to ***new Date();*** which will be the current date and time.

1. Create getters for **name**, **duration** and **startDate** as String
2. Every employee must have a working status. For this task you'll create a pubic **enum** called **Status** with the following values:

ACTIVE,

ONLEAVE,

PERMANENT,

CONTRACT

1. Let's create a class called **Trainer**
2. Create the following fieldsString **name**, Status **status**
3. Set the name and status fields using the Trainer's constructor
4. Every training course has a trainer assigned to it.   
   Add a new field to the *TrainingCourse* of type **Trainer**name thisfield **trainer**
5. Create getter and setters for **trainer** field
6. We can't run a training course without students. Before assigning students to a course, you must define a class called **Student** with the following fields:

String name, String stream     
(stream is either Development, Networking or Marketing. Could this be an enum?)

1. Set these fields using a constructor
2. Every training course must have one or more students.   
   We can discuss if this collection should be held by the TrainingCourse class or a separate class but for now, let's create an *ArrayList* of *Student* in the TrainingCourse class. Name this field **students**
3. Create a method called **addStudent** to add a students to the ArrayList  
   The method will have a parameter of type Student
4. Create a getter to get the names of all students attending a course.
5. Create a method called getDetails() which returns details of te training course, the names of all students and details of the trainer.
6. Test your code by writing code in the main() method to create a training course with a single trainer and a few students. You'll then call the getDetails() and check it's output.