# Analytical Programming

**Note: When completing these exercises all the code you write should appear within functions.**

## Question 1.

Please note that you should write a function called main. You will call each of the functions you write below from this main function.

(i)

Write a function called powerV1. When you call this function (from your main function) it should ask the user for a base number and a power number. It should then print out the result of raising the base number to the power of the second number. Sample output below:

```
Please enter base number: 3
Please enter power number: 2
The value 3 raised to the power of 2 is : 9
```

(ii)

You are going to write a variant of the program described in question (i). Your main function should ask the user to input two int values. The first will specify the base number and the second should specify power number. You will then called a function called powerV2. This function will take in two int values as parameters (the base and power numbers). The function will simply print the value of the base number raised to the power value.

(iii)

Write a variant of the function defined in part(ii) called powerV3. This function will take in two int values as parameters. The first will specify the base number and the second should specify power number. The function will calculate value of the base number raised to the power value and will **return** this value. Verify the function works correctly by storing the result returned from the powerV3 function in a variable and printing out the variable.

## Question 2.

(i)

Rewrite the calculator program from Week 3 to incorporate functions (you will find the solution to lab available in the Week 3 folder on Blackboard). The program you write should first ask the user for two separate numerical values. It should then give the user an option to perform one of four operations:  addition, subtraction, division or multiplication. Therefore, if the user selects multiplication then your program should print out the product of the two values.  You should write a different function for performing each operation. For example the sum function will take in two int numerical parameters and return the result. The following is sample output from this program.

```
Please enter a numerical value: 12
Please enter a numerical value: 10
Would you like to perform:
1: Addition
2. Subtraction
3. Multiplication
4 Division
> 3
Multiplication of 12 and 10 is 120
```

(ii)

Alter the program you wrote for part (i) so that it will repeatedly ask the user to enter two numbers and select an operation until they wish to exit. A user can exit by entering y/n when asked to select the operation they want to perform (See sample output on the next page).

```
Please enter a numerical value: 12
Please enter a numerical value: 10
Would you like to perform:
1: Addition
2. Subtraction
3. Multiplication
4 Division
> 3
Multiplication of 12 and 10 is 120
Would you like to perform another operation y\n? y
Please enter an integer value: 20
Please enter an integer value: 3
Would you like to perform:
1: Addition
2. Subtraction
3. Multiplication
4 Division
> 1
Addition of 20 and 2 is 23
Would you like to perform another operation y\n? n
Thank you for using our calculator.
```

**Question 3.**

Write a guessing game for a user. This program should initially generate a random number between 1 and 100.

It should then repeatedly ask the user to guess the random number.

Each time the user enters a guess the program should tell them that their guess was too high, too low or correct.

When the user finally guesses the correct number the program should tell the user how many guesses they made before arriving at the correct number.

Your program should make use of the following methods:

- generateRandomNumber. This function will generate a random number between 0 100 and return the result.
- askUser. This function will ask the user to enter a guess and will return the result
- checkGuess. This function will take in the users guess and the random number as parameters and will return True if the user entered the correct value and False otherwise.

```
Program has generated a random number:
Please enter your guess: 50
Too high
Please enter your guess: 25
Too low
Please enter your guess: 38
Correct. You made a total of 3 guesses.
```