**Unit 2: Selection and Iteration Constructs**

**Lab Exercises Part1**

Create a folder called unit2 labs on your OneDrive in folder Structures and Algorithms\Code to hold your lab solutions.

1. Write a program in a class called LeapYearTest that checks whether or not a year is a leap year.

A year is a leap year if it is exactly divisible by 4 (hint – remainder on integer division by 4 is 0). However if the year is a century year (exactly divisible by 100) then the year has to be exactly divisible by 400 to qualify as a leap year.

Use a Boolean variable that is set to true/false as appropriate and output the value of this variable.

Typical runs of the program (inputs are in bold):

year: **1996**

leap year: true

year: **1997**

leap year: false

year: **2000**

leap year: true

year: **2100**

leap year: false

2. (a) Write a program in a class called CalculatorTest that interacts with the user to get two integer values and then offers the options of addition, subtraction, multiplication, integer division (including remainder on integer division), floating point division or quit.

Typical runs of the program is (inputs are in bold):

first value: **14**

second value: **5**

1: add

2: subtract

3: multiply

4: integer division

5: floating point division

option: **1**

add: 14 + 5 = 19

first value: **14**

second value: **5**

1: add

2: subtract

3: multiply

4: integer division

5: floating point division

option: **2**

subtract: 14 - 5 = 9

etc.

(b) What happens if the second value is 0 and one of the division options is chosen?

(c) Update your program to fix this problem by preventing the erroneous operation(s) from taking place and outputting a suitable error message.

3. Write a program in a class called BalanceTest to maintain a bank account balance. The program should ask for an initial balance and then offer a menu of options to:

* make a deposit
* make a withdrawal
  + only valid if the new balance will not exceed the overdraft limit of £-1000
* add interest to the account at 5% of the balance
  + only valid if the account is not overdrawn
* add charges to the account at 10% of the balance
  + only valid if the account is overdrawn

After each transaction, display the new balance.

To simplify, the money amounts will be whole number values so use Integer as the type for the variables.

Typical runs of the program (inputs are in bold):

initial balance: **500**

select the add interest option

new balance: 525

initial balance: **500**

select the deduct charges option

no charges made as account is not overdrawn

new balance: 500

initial balance: **500**

select the deposit option

deposit amount: **75**

new balance: 575

initial balance: **500**

select the withdraw option

withdrawal amount: **2000**

withdrawal invalid as balance of -1500 would exceed the overdraft limit of -1000

new balance: 500

initial balance: **500**

select the withdraw option

withdrawal amount: **1000**

new balance: -500

initial balance: -**500**

select the add interest option

no interest added as account is overdrawn

new balance: -500

initial balance: -**500**

select the deduct charges option

new balance: -550