**Unit 2: Selection and Iteration Constructs**

**Lab Exercises**

Use the folder you have already created called unit2 labs\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 do … while loop with the repeat method of the Repeat class to allow the code to execute repeatedly. Use a Boolean variable that is set to true/false as appropriate and output the value of this variable.

A typical run of the program (inputs are in bold):

year: **1996**

leap year: true

continue(y/n)? **y**

year: **1997**

leap year: false

continue(y/n)? **y**

year: **2000**

leap year: true

continue(y/n)? **y**

year: **2100**

leap year: false

continue(y/n)? **n**

2. Combine the results of self-assessments 1 and 7 to create a program in a class called WeeklyWageTest that calculates a weekly wage based on the number of hours worked. The program should be able to deal with a number of employees (use the repeat method from the Repeat class to control the program). The program should output the weekly hours worked and the wage.

For inputs of 6 6 6 6 6 0, output should be: hours worked: 30 wage 450

For inputs of 7 7 7 7 7 0, output should be: hours worked: 35 wage 525

For inputs of 7 7 7 7 7 5, output should be: hours worked: 40 wage 650

3. (a) Write a program in a class called CalculatorTest that interacts with the user to get two integer values and then repeatedly offers the options of addition, subtraction, multiplication, integer division (including remainder on integer division), floating point division or quit. The program should terminate when the quit option is selected.

A typical run of the program is (inputs are in bold):

first value: **14**

second value: **5**

0: quit

1: add

2: subtract

3: multiply

4: integer division

5: floating point division

option: **1**

add: 14 + 5 = 19

0: quit

1: add

2: subtract

3: multiply

4: integer division

5: floating point division

option: **2**

subtract: 14 - 5 = 9

0: quit

1: add

2: subtract

3: multiply

4: integer division

5: floating point division

option: **0**

(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.

4. You are to write a program in a class called ExamMarksTest that outputs the result that students have obtained in an exam. The exam has 4 questions each marked out of 25. The pass mark is 40 the merit mark is 70.

step 1: Write the code to get the mark for each question from input and then output the overall exam mark.

step 2: Write the code to output whether or not the student has passed with merit, passed or failed.

step 3: The exam mark for each question should be in the range 0 to 25. Write code that repeatedly asks for the mark for each question until a valid mark is entered (for test purposes assume that only whole number values will be entered but could be outside of the expected range).

step 4: Write the code to allow the program to deal with a number of students (use the repeat method from the Repeat class to control the program).

A typical run of the program is (inputs are in bold):

mark for question 1: **15**

mark for question 2: **18**

mark for question 3: **30**

mark for question 3: **40**

mark for question 3: **20**

mark for question 4: **-5**

mark for question 4: **15**

exam mark: 68

student has passed

continue(y/n)? **y**

mark for question 1: **22**

mark for question 2: **18**

mark for question 3: **20**

mark for question 4: **15**

exam mark: 75

student has passed with merit

continue(y/n)? **y**

mark for question 1: **10**

mark for question 2: **8**

mark for question 3: **15**

mark for question 4: **3**

exam mark: 36

student has failed

continue(y/n)? **n**

5. 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.

A typical run of the program (inputs are in bold):

initial balance: **500**

select the add interest option

new balance: 525

select the deduct charges option

no charges made as account is not overdrawn

new balance: 525

select the deposit option

deposit amount: **75**

new balance: 600

select the withdraw option

withdrawal amount: **2000**

withdrawal invalid as balance of -1400 would exceed the overdraft limit of -1000

new balance: 600

select the withdraw option

withdrawal amount: **1000**

new balance: -400

select the add interest option

no interest added as account is overdrawn

new balance: -400

select the deduct charges option

new balance: -440