THE UNIVERSITY OF BUCKINGHAM

SCHOOL OF SCIENCE

Exam Title:	**PROBLEM SOLVING AND PROGRAMMING 1**

SPRING 2017

EXAMINATION FOR THE DEGREE PROGRAMMES IN

Computing, October 2016 entry and January 2017 Entry

Exam Code:	20172AIAP013AWP
TIME ALLOWED:	**2 HOURS**
MATERIALS PERMITTED:	None
MATERIALS PROVIDED:	ExamReferencePSP on P drive

INSTRUCTIONS:

1. Answer ALL questions.
2. Questions 1 to 3 require only the practical programming work whereas Question 4 requires both written work in the answer book and the practical programming work.
3. You must save all your practical work on the designated folder on the desktop of your computer.
4. You must organize your answers into projects: one project for each question.
5. **Do not** provide more than one answer to each question.
6. You may use the file “ExamReferencePSP” on the P drive for a quick reference of C language syntax.

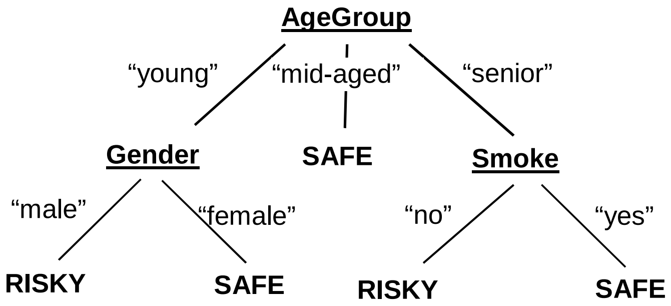
Question 1 (15 Marks)

A vending machine sells small food items. To buy an item, the user inserts an amount of payment equal or more than the item price. The machine should then return the item with the right amount of change in terms of the fewest number of coins.

Write a program that receives the item price and the amount paid (both in pound sterling), calculates the amount of change, and displays the fewest number of coins for the change. The types of coins considered are 10p, 5p and 1p.

Question 2 (15 Marks)

Insurance companies often assess the risk factor of a client purchasing their products. Write a program that receives necessary data inputs, applies the following decision scheme to the data inputs and classifies a client as being either SAFE or RISKY:



Question 3 (20 Marks)

A bank saving scheme allows a client to save a certain amount of money over a number of years at a fixed annual interest rate.

Write a program that receives necessary inputs from the user and displays the final amount of saving.

Question 4 (50 marks)

This question is concerned with developing and testing a library of functions. **The question requires you to write your solution for each function in either pseudo code or flowchart in the answer book, construct the program components, and then test if the program components work.**

Known as “digitLib”, the library should provide three functions all of which work on the digits of a given integer. The purpose of each function is listed as follows:

- *digitCount* that counts the number of digits of a given positive integer. For instance, for integer 829, the function will return 3.
  (15 marks)

- *digitSum* that adds the digits of a given positive integer into a total sum. For instance, for integer 829, the function will return 19 (8 + 2 + 9 = 19).
  (15 marks)

- *digitK* that returns the k^{th} digit of a given positive integer. For instance, for integer 829, if k = 1, the function will return the last digit 9. If k = 2, the function will return the middle digit 2. If k = 3, the function will return the third digit 8. If k has any number greater than 3, the function will return 0.
  (15 marks)

You should follow the normal steps of developing library functions, i.e. function declarations in digitLib.h, function definitions in digitLib.cpp, and function calls in the main program.
(5 marks)

Please note: recursive solutions receive more credits than non-recursive solutions.

--- END OF PAPER ---