COMP101

Introduction to Programming 2019-20

Assignment-03

Issue Date: **Monday 21st October 2019**

Submission Date: **Monday 28th October (12:00 noon)**

Summary:

Assignment-03 is worth 13% of the total marks for COMP101.
The assignment uses sequencing, selection, iteration and I/O control and requires computational thinking and pseudocode.
You must submit an attempt at this assignment else a fail grade for the module will be awarded.

Guidance:
Assessment is based on design, clarity, accuracy and appropriate use of code, testing and documentation.

Deadline and submission details:
a) Submit **one** .py file with filename in format of:

familyName_givenName-CA03.py e.g. Smith_John-CA03.py

Within the code, the first 3 lines should be comment lines as follows:
#Your University id and filename
#Month and Year of coding
#Brief description of the problem solved

b) Submit one .pdf file containing your test table and pseudocode.
Use the testing template for your evidence of testing

c) Your documents are to be submitted electronically via the department submission server at https://sam.csc.liv.ac.uk/COMP/Submissions.pl

Earlier submission is possible, but any submission after the deadline attracts the standard lateness penalties - see http://www.csc.liv.ac.uk/department/regulations/practical.html

Plagiarism and collusion guidelines will apply throughout the assignment submission

COMP101 Assignment-03 2019-20

Assessment Information

Assignment Number	03 (of 07)
Weighting	13%
Assignment Circulated	As front page
Deadline	As front page
Submission Mode	e-submission
Learning outcome assessed	LO1: Identify principles and practice of using high level programming constructs to solve a problem LO4: Use a suitable Integrated Development Environment to carry out implementation, interpretation/compilation, testing and execution LO6: Design and apply effective test cases LO7: Develop debugging skills to correct a program
Purpose of assessment Marking criteria	Assessment of using sequence, selection and iteration constructs and in computational thinking and pseudocode Total marks over seven questions as a percentage
Submission necessary in order to satisfy module requirements?	Yes Assignments are not marked anonymously
Late Submission Penalty	Standard UoL Policy.

Problem Specification:
Calculate the age of a cat in human years.

Only use the equivalence table below – no alternatives please

Life stage	Age of cat	Human equivalent
Kitten	1 month	1 year
birth to 7 months	2 months	2 years
	3 months	3 years
	4 months	8 years
	5 months	9 years
	6 months	10 years
Overlap is okay	7 months	11 years
Overlap is okay	8 months	11 years
Junior	12 months	15 years
12 months to 24 months	18 months	21 years
	24 months	27 years
Prime	3 years	28 years
3 years to 6 years	4 years	32 years
	5 years	36 years
	6 years	40 years
Mature	7 years	44 years
7 years to 10 years	8 years	48 years
	9 years	52 years
	10 years	56 years
Senior	11 years	60 years
11 years to 14 years	12 years	64 years
	13 years	68 years
	14 years	72 years
Geriatric	15 years	76 years
15 years to 18 years	16 years	80 years
	17 years	84 years
	18 years	88 years

Present the user with a main menu to allow input from three options:

a) The user can select to enter an age for a kitten/junior life stage:
Integer input only – no floats
Inform the user which life-stage their cat is at and what their age is in human years.
Return the user to the main menu – the program only ends when X is input

or

b) the user can select to enter an age for prime, mature, senior, geriatric life stages
Integer input only – no floats
Inform the user which life-stage their cat is at and what their age is in human years.
Return the user to the main menu – the program only ends when X is entered

or

c) the user can exit the menu (and the program) by pressing option 'X'
When option 'X' is entered, terminate the program.

Appropriate use of iteration will control the menu and validation

Mark scheme

Analysis and Design – pseudocode is required	20%
Implementation	50%
Testing	15%
Effective algorithm giving efficient statements	15%

The mark scheme looks for:

i) A sequence, selection and iteration approach to solving the problem, with effective use of selection.
An accurate solution that uses an algorithm to avoid an excessive number of selection and/or compound Boolean statements will score higher than a solution that over-uses such statements

ii) Efficient use of variables to handle the input data and the use of theses variables to make the code clear and readable, thus aiding maintenance and debugging

iii) Appropriate use of output techniques that are of benefit to the user

iv) Use in-line comments sparingly, but effectively
The algorithm chosen should be suitably explained in the commentary

v) Appropriate testing to determine accuracy and/or problems (which may be documented in the test table comments column)

Do not change the problem specification. Assessment is based only on the equivalence table as given - this keeps the testing and marking consistent