

CSC1010H Tutorial 3

Selection, loops

Introduction

Learning outcomes

Skills

- Be able to construct and reason about Python solutions to programming problems involving selection and/or iteration.
- Including,
 - Constructing and reasoning about compound Boolean expressions involving the use of python comparison operators.
 - Constructing and reasoning about data selection expressions that use of string slicing and casting.

Knowledge

- The forms of Python if, if-else and if-elif statements.
- Python comparison operators, the Boolean operators 'and', 'not' and 'or', and operator precedence.
- The form of the Python for loop, and the use of break and continue statements.
- The role of the range function in for loops.
- The distinction between definite and indefinite loop structures.
- Forms and possible uses of string slice and casting operations.

Question 1 [20 marks]

Write a program called `'monthlycalendar.py'` that asks the user to enter a month number (where 1 is January) and a year. It will output the number of days in that month for that year.

Your program should ensure that values input by the user are positive and, in the case of the month number, not greater than 12.

Sample I/O:

```
Enter number of month (1 is January):  
1  
Enter year:  
2008  
In the year 2008, January has 31 days.
```

(User input in **bold**.)

Sample I/O:

```
Enter number of month (1 is January):  
2  
Enter year:  
2008  
In the year 2008, February has 29 days.
```

(User input in **bold**.)

Sample I/O:

Enter number of month (1 is January):

-2

Enter year:

2008

Sorry, invalid input(s).

(User input in **bold**.)

Hint: A year is a leap year if divisible by 4 and not 100 unless it's divisible by 400.

Question Two [25 marks]

Write a program called 'finalmark.py' which calculates the final mark of a first-year university Computer Science student, and indicates what grade pass the student achieved.

The program will allow the user to enter the following information for a student: the student's name and student number, their percentage mark for programming assignments, their percentage mark for class tests, and their percentage mark for the final exam.

The Final mark is calculated with the marks weighted in the following way:

- programming assignments count 30%
- class tests count 20%
- final exam counts 50%

Once the Final mark is calculated, assign a grade to the student, using the following criteria:

<i>Final mark (m).</i>	<i>Grade</i>
$75\% \leq m \leq 100\%$	1st
$60\% \leq m < 75\%$	2nd
$50\% \leq m < 60\%$	3rd
$0\% \leq m < 50$	Fail

Assume for the following sample I/O that we have a student, Jet Li, with student number lxxjet001, who has obtained 50% for programming assignments, 50% for class tests and 50% for the final exam.

```
*** Final Mark Calculator ***
```

```
Enter the student's name:
```

```
Jet Li
```

```
Enter the student number:
```

```
lxxjet001
```

```
Enter the programming assignment mark:
```

```
50%
```

```
Enter the class test mark:
```

```
50%
```

```
Enter the final exam mark:
```

```
50%
```

```
Results:
```

```
Student Jet-Li (number lxxjet001) has a final mark of 50%.
```

```
This mark is graded as a 3rd.
```

(User input in **bold**.)

You may assume that input marks are all integer values. You should round the final mark to the nearest integer.

Hint: The expression `round(5.3)` is 5 and `round(5.7)` is 6.

Question 3 [30 marks]

Write a program called `'imperialmetric.py'` that computes and displays a conversion table for feet and inches to metres. The program should ask the user to enter the range of values that the table will hold.

Here is an example of what should be output when the program runs:

```
Enter the minimum number of feet (not less than 0):
8
Enter the maximum number of feet (not more than 30):
11
|      0"      1"      2"      3"      4"      5"      6"      7"      8"      9"     10"     11"
8' | 2.438 2.464 2.489 2.515 2.540 2.565 2.591 2.616 2.642 2.667 2.692 2.718
9' | 2.743 2.769 2.794 2.819 2.845 2.870 2.896 2.921 2.946 2.972 2.997 3.023
10'| 3.048 3.073 3.099 3.124 3.150 3.175 3.200 3.226 3.251 3.277 3.302 3.327
11'| 3.353 3.378 3.404 3.429 3.454 3.480 3.505 3.531 3.556 3.581 3.607 3.632
```

We recommend the following formula:

```
metres = (feet*12+inches)*0.0254;
```

Hint:

- You are going to need a loop within a loop.
- If you are stuck, try developing your program in parts.

Try, for instance:

- Developing a program that computes metres from feet:

```
Enter the minimum number of feet (not less than 0):
5
Enter the maximum number of feet (not more than 99):
10
5' | 1.52m
6' | 1.83m
7' | 2.13m
8' | 2.44m
9' | 2.74m
10'| 3.05m
```

- Developing a program that computes metres from inches:

```
Enter the minimum number of feet (not less than 0):
5
Enter the maximum number of feet (not more than 99):
10
```

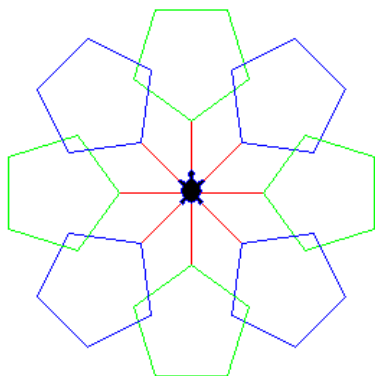
```

5' |
6' |
7' |
8' |
9' |
10' |

```

Question 4 [25 marks]

Consider the following flower pattern:



It consists of 8 petals (polygons) on red stems (spokes). The petal colour alternates between green and blue.

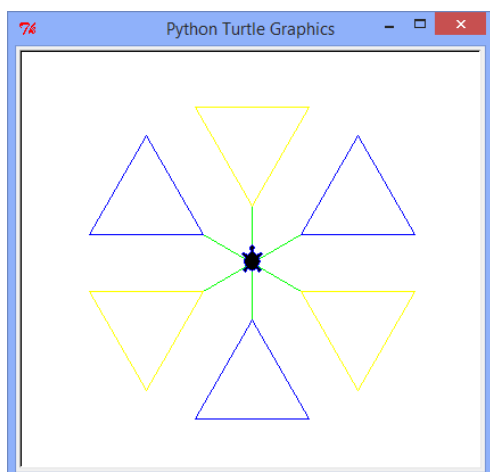
Write a Python Turtle program called *flowerpattern.py* which allows the user to draw patterns of this sort. The program will ask the user to enter the number of petals, the length of stem, the sort of petal (triangle, square, or pentagon), the length of petal side, and the colours.

Here is a sample program execution:

```

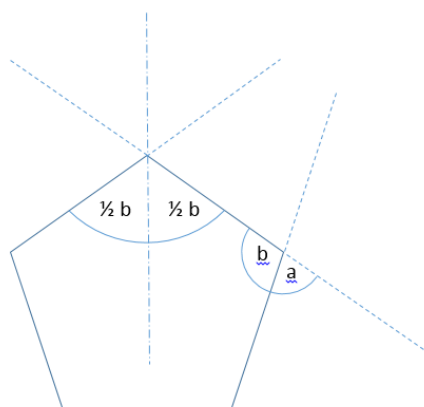
Enter the number of petals (must be an even number): 6
Enter the length of stem: 50
Enter the petal type (triangle, square, or pentagon): triangle
Enter the length of side: 100
Enter the stem colour: green
Enter the first petal colour: blue
Enter the second petal colour: yellow

```



Hint:

- As with the previous exercise, consider breaking the problem down into simpler tasks and then building and combining solutions. For example,
 - Write a program that obtains the required inputs (and translates petal type to the required number of polygon sides).
 - Extend the program to draw the stems/spokes.
 - Write a program that asks the user to enter 3, 4, or 5, and draws a polygon with that many sides.
- How do you centre a petal on its stalk? The turtle will turn a certain amount clockwise or a certain amount anti clockwise before drawing the polygon (depending on whether it is drawn anti clockwise or clockwise):



Facts: (i) $a = 360 / (\text{number of sides})$ and (ii) $a + b = 180$.

Marking and Submission

Submit the files `monthlycalendar.py`, `finalmark.py`, `imperialmetric.py` and `flowerpattern.py` contained within a single .ZIP folder to the automatic marker. The zipped folder should have the following naming convention:

yourstudentnumber.zip

Marking Guide

Exercise One – Monthly Calendar	20
Exercise Two – Final Mark	25
Exercise Three – Imperial to Metric conversion	30
Exercise Four – Python Turtle Polygons	25
Total	100

Note that exercise 4, `flowerpattern.py`, will be marked manually.