KIT100 Programming Preparation

Tutorial Four – Week 5

Today's Flow

- Tutorial Tasks
  - to practice reading through code
  - to review good programming principles
  - to become familiar with using operators
  - to be able to engage in decision making with Python
- Walk through Portfolio Tasks PP5.1 & PP5.2

Python Code Reading

You are given this Python source code.

```
x = int(input("Please enter a number: "))
print("%s minutes is approximately %s years and %s days" % (x,(x//(24*60))//365,(x//(24*60))%365))
```

- Do you know the objective of this program?
- Do you think it is easy to read?
- Do you think it is easy to modify the logic?

Python Code Reading

How about this one? Is it easier?

Please spend **5 minutes**. Could you able to read it **line by line**, and understand the **overall logic flow**?

```
1   '''
2   Minutes to days to years conversion
3   Author: David Herbert
4   Version 1.1
5   Date: March 2016
6   Purpose: To convert an entered int read as minutes to days and years
7   '''
8   # Obtain input
9   minutes = int(input("Enter the number of minutes: "))
10
11  numberOfDays = minutes // (24 * 60)
12  numberOfYears = numberOfDays // 365
13  numberOfActualDays = numberOfDays % 365
14
15  # Display results
16  print("%s minutes is approximately %s years and %s days" \
17     % (minutes, numberOfYears, numberOfActualDays))
```

?

?

?

?

?

Any **variables**?

Any **operators**?

Objective of this program: ?

Python Code Reading

You are given this Python source code. Could you able to read it **line by line**, and understand the **overall logic flow**?

```
1   '''
2   Minutes to days to years conversion
3   Author: David Herbert
4   Version 1.1
5   Date: March 2016
6   Purpose: To convert an entered int read as minutes to days and years
7   '''
8   # Obtain input
9   minutes = int(input("Enter the number of minutes: "))
10
11  numberOfDays = minutes // (24 * 60)
12  numberOfYears = numberOfDays // 365
13  numberOfActualDays = numberOfDays % 365
14
15  # Display results
16  print("%s minutes is approximately %s years and %s days" \
17     % (minutes, numberOfYears, numberOfActualDays))
```

Get minutes from user input

Calculate number of days by dividing and returning the integer value of the quotient.

Calculate number of years by dividing the result from number of days.

Divides and returns the value of the remainder

Output the result

Variables

Operators

=	Assignment operator
*	Multiplication operator
//	Floor division operator
%	Modulus operator

Objective of this program: To convert an entered integer read as minutes to days and years

Good Programming Principles

"Programs must be written for people to read, and only incidentally for machines to execute."
– Harold Abelson, Structure and Interpretation of Computer Programs

"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."
– Martin Fowler 2008

It is important to know the rules of programming, which is **Good Programming Principles**!

- **Line Length**
  - Limit your source code lines to a maximum **79** characters
- **Blank Lines (White spaces)**
  - Provide enough space to distinguish between different logic
- **Library Import**
  - Should always at the top of your source code and in separate lines.

- **String Quotes**
  - Single quote or Double quote are both OK in Python. But please stick into one only in your source code.
- **Naming Conventions**
  - Meaningful naming!
  - lower case with underscore for function_name
  - camelCase for variableName / lower_case_for_variable_name
  - UPPERCASE FOR CONSTANTS

Good Programming Principles

Which one follows good programming principles?

1)

```
if (width == 0 and height == 0 and
    color == 'red'and
    emphasis == 'strong' or highlight > 100):

    print ("Ohh...It works!")
```

2)

```
if (width == 0 and height == 0 and color == 'red' and emphasis == 'strong' or highlight > 100):

    print ("Ohh...It works!")
```

Good Programming Principles

Which one follows good programming principles?

1)

```
if (width == 0 and height == 0 and
    color == 'red'and
    emphasis == 'strong' or highlight > 100):

    print ("Ohh...It works!")
```

```
if (width == 0 and height == 0 and color == 'red' and emphasis == 'strong' or highlight > 100):

    print ("Ohh...It works!")
```

Good Programming Principles

Which is the correct use of library import?

1)
```
import os
import sys
```

2)
```
import sys, os
```

Good Programming Principles

Which is the correct use of library import?

1)

```
import os
import sys
```

In separate lines.

Please also remember to import libraries at the top of your source code.

```
import sys, os
```

Good Programming Principles

Which is the correct use of whitespace around operators?

1)

```
i= i + 1
submitted += 1
x = x*2 - 1
hypot2 = x*x + y*y
c = (a+b) * (a-b)
```

2)

```
i=i+1
submitted +=1
x = x * 2- 1
hypot2 = x * x + y * y
c = (a + b) * (a - b)
```

Good Programming Principles

Which is the correct use of whitespace around operators?

1)

```
i= i + 1
submitted += 1
x = x*2 - 1
hypot2 = x*x + y*y
c = (a+b) * (a-b)
```

if operators with different priorities are used, consider adding whitespace around the operators with the lowest priority(ies).

```
i=i+1
submitted +=1
x = x * 2- 1
hypot2 = x * x + y * y
c = (a + b) * (a - b)
```

Decision Making

Program flow

Run only when `conditionA` is **True**

```
if (conditionA):
    program statement
elif (conditionB):
    program statement
else:
    program statement
```

4 space characters

Run only when `conditionB` is **True**

Program flow

```
1  weight = float(input("Please input your weight in kg: "))
2  height = float(input("Please input your height in meters: "))
3  bmi = weight / ( height * height )
4
5  print("Your BMI is:", bmi)
6  if (bmi < 18.5):
7      print("You are underweight.")
8  elif (bmi > 25):
9      print("You are overweight.")
10 else:
11     print("You are in normal weight.")
```

```
Please input your weight in kg: 75
Please input your height in meters: 1.7
Your BMI is: 25.95155709342561
You are overweight.
```

Review Programming Principles

Please spend **20 minutes** to complete the table on the right.

You are given the following program fragment:

```
c = 'Y'
s = "whatever"
i = 34
b = False
```

Could they be used as the if conditions?
If **yes**, what will be the result?
If **no**, why not?

Expression	Valid?	Result/Reason
`c=!'2'`	No	operator =! Is invalid
`c<'y'`	Yes	`True`
`s=="whatever"`		
`s==s`		
`input()`		
`b`		
`b==False`		
`b!=True`		
`b=False`		
`len(s)==i`		
`"cat" + "dog"`		

Review Programming Principles

Please spend **20 minutes** to complete the table on the right.

You are given the following program fragment:

```
c = 'Y'
s = "whatever"
i = 34
b = False
```

Could they be used as the if conditions?
If **yes**, what will be the result?
If **no**, why not?

Expression	Valid?	Result/Reason
`c=!'2'`	No	operator =! Is invalid
`c<'y'`	Yes	`True`
`s=="whatever"`	Yes	`True`
`s==s`	Yes	`True`
`input()`	Yes	`True` when user input something `False` when user input nothing
`b`	Yes	`False`
`b==False`	Yes	`True`
`b!=True`	Yes	`True`
`b=False`	No	= operator is assignment operator, which cannot be used in if conditions
`len(s)==i`	Yes	`False`
`"cat" + "dog"`	Yes	`True`

Review Programming Principles

Well plan on the if conditions is important when doing the development.

Please spend **10 minutes** to complete the questions on the right.

Assume your program starts as follow,

```
name = input("Please enter your title and first name: ")
```

Assuming the user has been asked for their name and has entered this into a character string variable name, for each of the following scenarios, identify which **conditional statement** should be used:

i. Display their name prefaced with the word "Hello".
☐ if ☐ if-else ☐ if-elif ☐ nested if ☑ no conditional required

ii. Display the phrase "Thank you sir" if the name variable begins "Mr ".
☑ if ☐ if-else ☐ if-elif ☐ nested if ☐ no conditional required

iii. Display the phrase "an odd name…" if the length of the name variable is odd and "an even name…" if the length is not odd.
☐ if ☐ if-else ☐ if-elif ☐ nested if ☐ no conditional required

iv. Display the word "two" if there are two characters in the name variable, "three" if the name variable is three characters long, "four" if of length four, and so on, up to "nine". Only one message should be displayed.
☐ if ☐ if-else ☐ if-elif ☐ nested if ☐ no conditional required

v. Display the message "Not a valid name" if the user's name starts with a number.
☐ if ☐ if-else ☐ if-elif ☐ nested if ☐ no conditional required

Review Programming Principles

Well plan on the if conditions is important when doing the development.

Please spend **10 minutes** to complete the questions on the right.

Assume your program starts as follow,

```
name = input("Please enter your title and first name: ")
```

Please also spend some times to complete the python script according to the questions on the right.

Assuming the user has been asked for their name and has entered this into a character string variable name, for each of the following scenarios, identify which **conditional statement** should be used:

i. Display their name prefaced with the word "Hello".
☐ if ☐ if-else ☐ if-elif ☐ nested if ☑ no conditional required

ii. Display the phrase "Thank you sir" if the name variable begins "Mr ".
☑ if ☐ if-else ☐ if-elif ☐ nested if ☐ no conditional required

iii. Display the phrase "an odd name…" if the length of the name variable is odd and "an even name…" if the length is not odd.
☐ if ☑ if-else ☐ if-elif ☐ nested if ☐ no conditional required

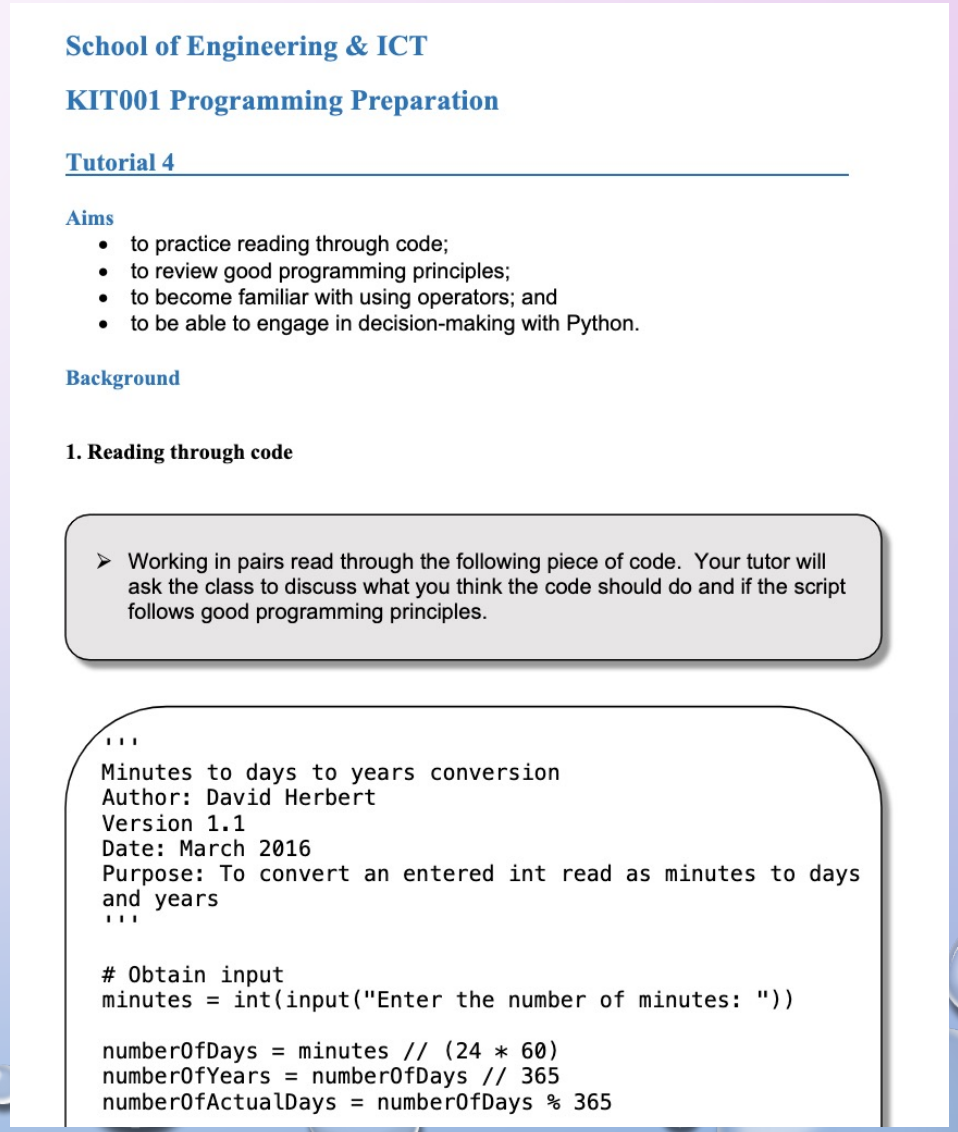
iv. Display the word "two" if there are two characters in the name variable, "three" if the name variable is three characters long, "four" if of length four, and so on, up to "nine". Only one message should be displayed.
☐ if ☐ if-else ☑ if-elif ☐ nested if ☐ no conditional required

v. Display the message "Not a valid name" if the user's name starts with a number.
☑ if ☐ if-else ☐ if-elif ☐ nested if ☐ no conditional required

Review and Practise

- Open **KIT100 Tutorial Four.pdf** on MyLO

- The items in this section are from tutorials from previous years. Feel free to test your comprehension of the unit so far and answer questions.



School of Engineering & ICT

KIT001 Programming Preparation

Tutorial 4

Aims

- to practice reading through code;
- to review good programming principles;
- to become familiar with using operators; and
- to be able to engage in decision-making with Python.

Background

1. Reading through code

➢ Working in pairs read through the following piece of code. Your tutor will ask the class to discuss what you think the code should do and if the script follows good programming principles.

```
'''
Minutes to days to years conversion
Author: David Herbert
Version 1.1
Date: March 2016
Purpose: To convert an entered int read as minutes to days
and years
'''

# Obtain input
minutes = int(input("Enter the number of minutes: "))

numberOfDays = minutes // (24 * 60)
numberOfYears = numberOfDays // 365
numberOfActualDays = numberOfDays % 365
```

Repetitive Tasks

Imagine you have to print multiplication table for your kids:

Which one do you prefer?

1)

```
print('1 x 2 = 2')
print('2 x 2 = 4')
print('3 x 2 = 6')
print('4 x 2 = 8')
print('5 x 2 = 10')
print('6 x 2 = 12')
print('7 x 2 = 14')
print('8 x 2 = 16')
print('9 x 2 = 18')
print('10 x 2 = 20')
print('11 x 2 = 22')
print('12 x 2 = 24')
```

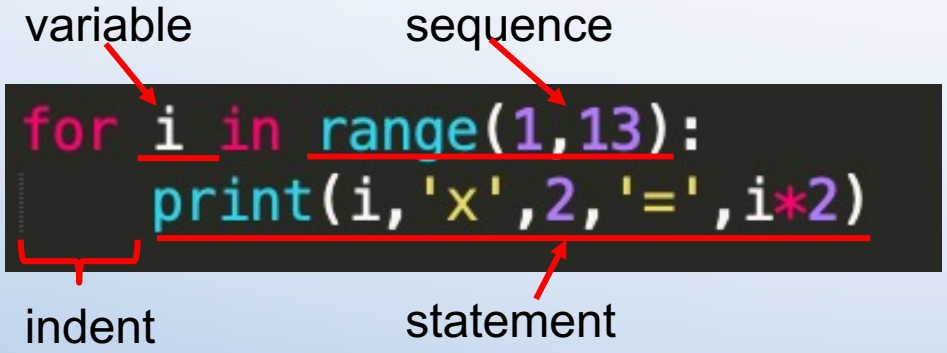
2)

```
for i in range(1,13):
    print(i,'x',2,'=',i*2)
```

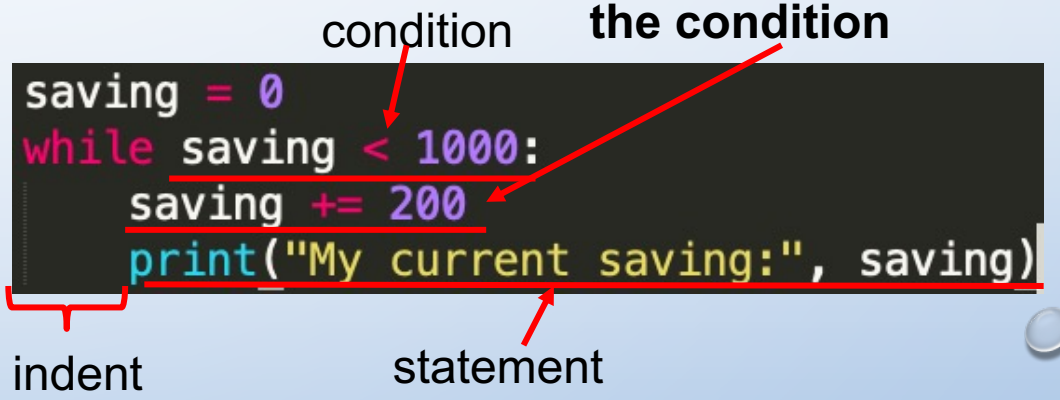
Repetitive Tasks

for loops executes a fixed number of times, and you know the number of times it will execute before the loop even begins.

while loops executes statements repeatedly as long as a **condition remains true**.



IMPORTANT:
Statement to change the condition



Portfolio Tasks

- **5.1PP Looping (for)**

Description:

Different people can climb steps at different rates (number of climbing-steps per minute). A tourist company has asked you to create a program that tourists can use to determine how many steps they can climb in a certain amount of time to get to a scenic lookout which is 1000 steps from where they start.

Task:

Write a Python program that **asks the user** to enter their stair-climbing rate (in steps per minute), followed by a time duration (in minutes).

The program will then display a heading, followed by several rows of data (using a **for** loop), with each row (suitably aligned to the heading) displaying the elapsed minute (starting at 1), followed by the cumulative number of steps climbed. The last row *time* should match the duration entered by the user.

Your program should finally indicate which minute (if at all) they would reach the lookout situated 1000 steps above that starting point. If the lookout was not reached, a suitable message should be displayed.

Hint:

Use the *range()* function.

Example output:

```
Please enter your stair step rate in steps per minute: 250
Please enter the length of time in minutes to display: 6

Time Step total
---------------
  1   250
  2   500
  3   750
  4  1000
  5  1250
  6  1500
You reached the lookout at 1000 steps in 4.0 minutes
```

Portfolio Tasks

- **5.2PP Looping (while)**

Description:

A statistician wants a program to allow her to enter multiple numbers (integers) and show their product (multiply all inputs) when all numbers have been entered.

Task:

Write a Python program that first tells the user what it does, and then **asks the user** to enter a series of integers (positive or negative). The user should enter zero to signal the end of the series. After all the numbers have been entered, the program should display their product and the count of how many numbers were entered.

Example output:

```
I multiply your numbers together.
Enter your next number, 0 to finish: 2
Enter your next number, 0 to finish: 3
Enter your next number, 0 to finish: 4
Enter your next number, 0 to finish: 0

The product of your numbers is 24
You entered 3 numbers
```