KIT100 Programming Preparation

Tutorial Two – Week 3

Tutor Team

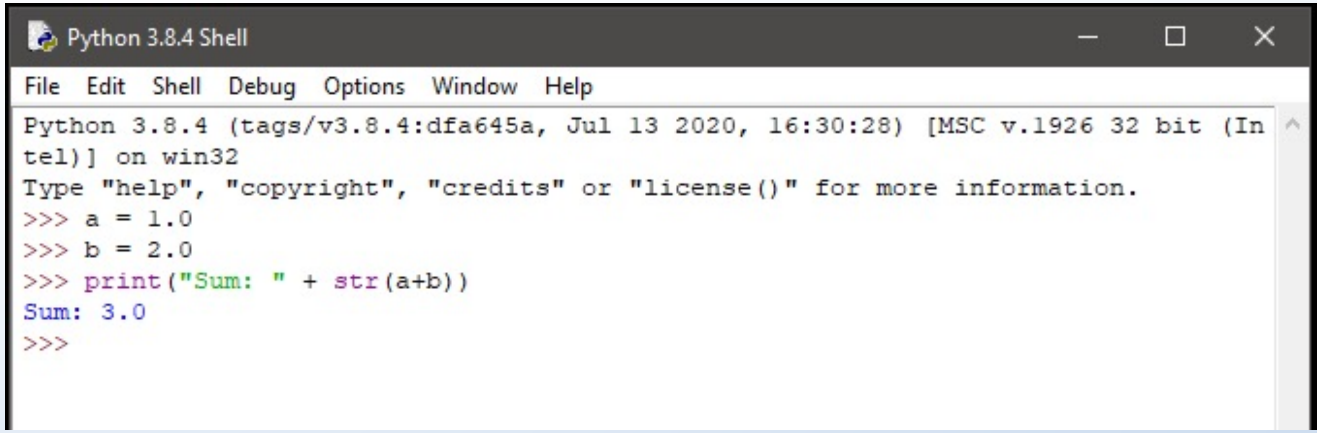
- Your Wednesday Tutor: Dr. Son Tran
  - Email: sn.tran@utas.edu.au
  - Consultation Hour: (to be confirmed)
- Your Thursday Tutor: Jamal Maktoubian
  - Email: jamal.maktoubian@utas.edu.au
  - Consultation Hour: (to be confirmed)

Today’s Flow

- Execute Python in IDLE
- Create your first Python program
- Experience on how to debug
- Take Attendance
- Your time to work on the tutorial tasks

Execute Python in IDLE

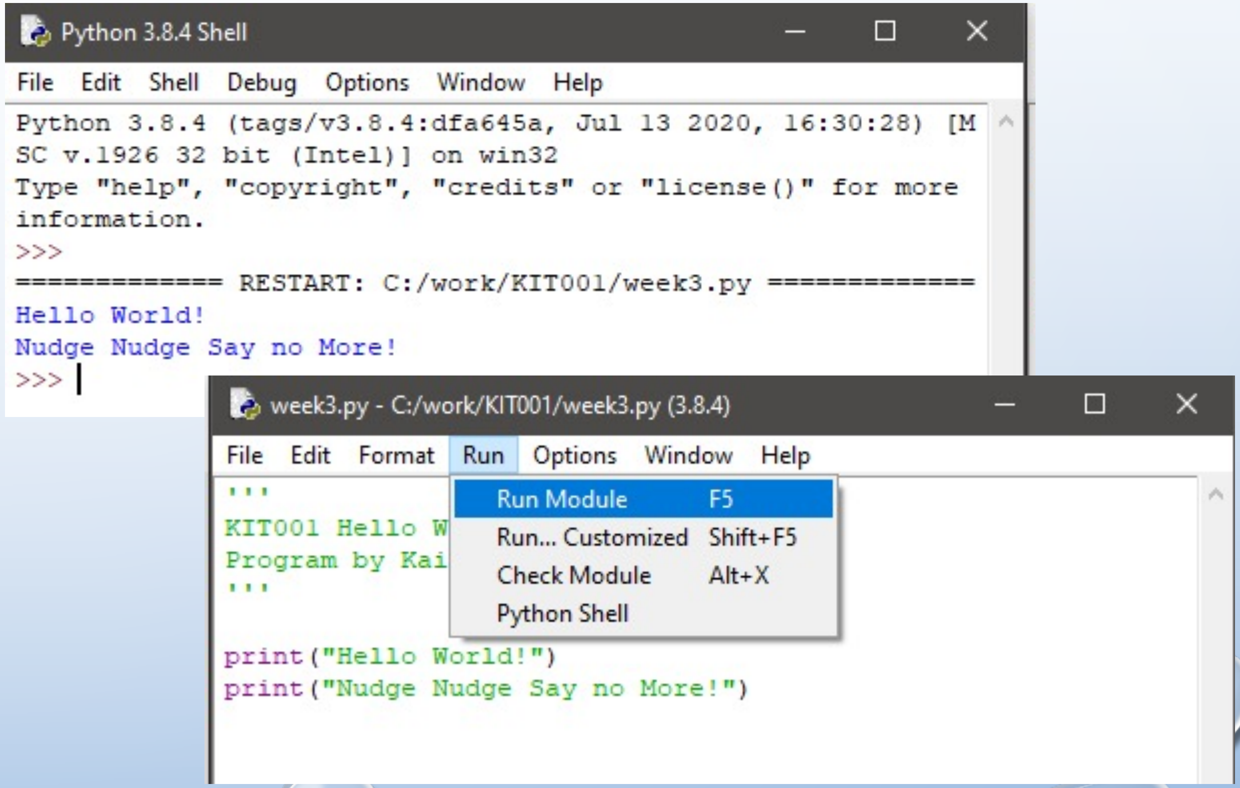
- You can run Python statement step by step in IDLE.
- It is convenient, but could be messy when your program is getting complicated.



```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:30:28) [MSC v.1926 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 1.0
>>> b = 2.0
>>> print("Sum: " + str(a+b))
Sum: 3.0
>>>
```

Create your first Python Program

- Use any text editor or build in text editor in IDLE.
- Remember to save the text file as **.py** for your Python program.
- Click Run > Run Module to execute your program.


Python 3.8.4 Shell
File Edit Shell Debug Options Window Help
week3.py - C:/work/KIT001/week3.py (3.8.4)
File Edit Format Run Options Window Help
Run Module F5
Run... Customized Shift+F5
Check Module Alt+X
Python Shell

```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:30:28) [M
SC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
============= RESTART: C:/work/KIT001/week3.py =============
Hello World!
Nudge Nudge Say no More!
>>>
```

```
'''
KIT001 Hello W
Program by Kai
'''

print("Hello World!")
print("Nudge Nudge Say no More!")
```

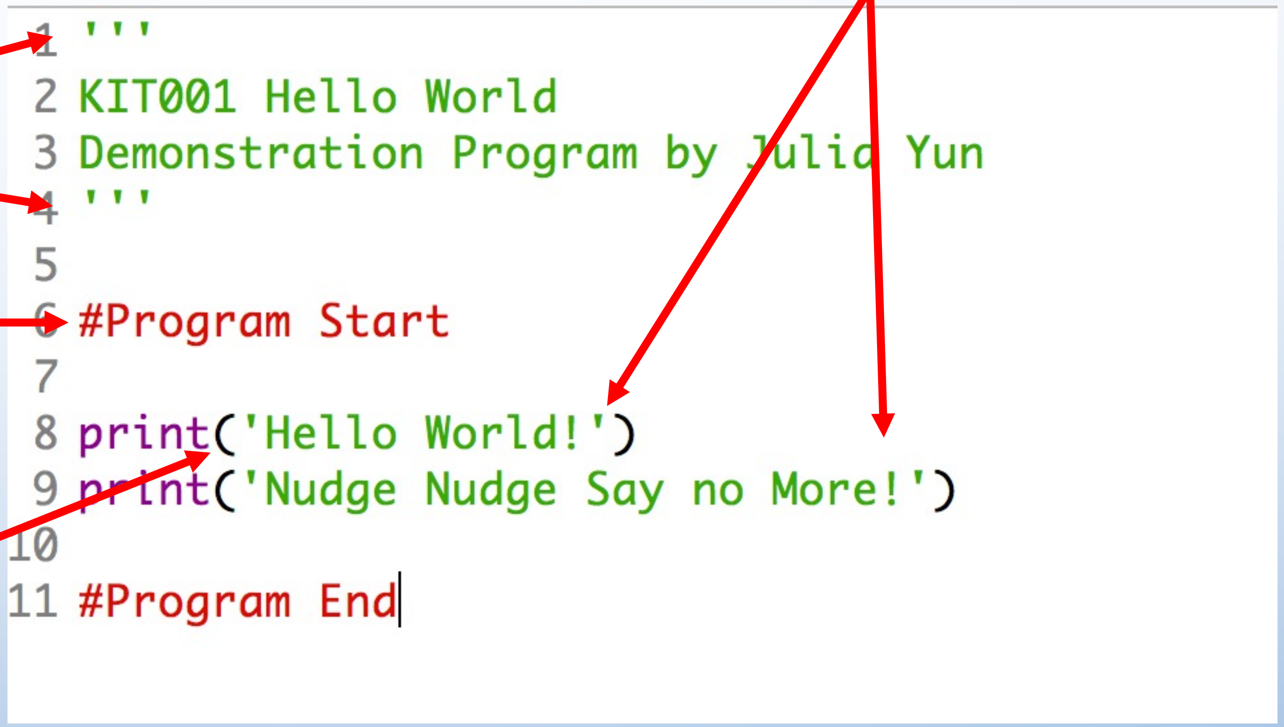
Python Syntax

''' (Triple single quotation marks)
Enclosing a multiple line comment

#
Single line comment

()
Function call

' ' or " "
Enclose a string



```
'''
KIT001 Hello World
Demonstration Program by Julia Yun
'''

#Program Start

print('Hello World!')
print('Nudge Nudge Say no More!')

#Program End
```

Python Syntax

CONSTANT

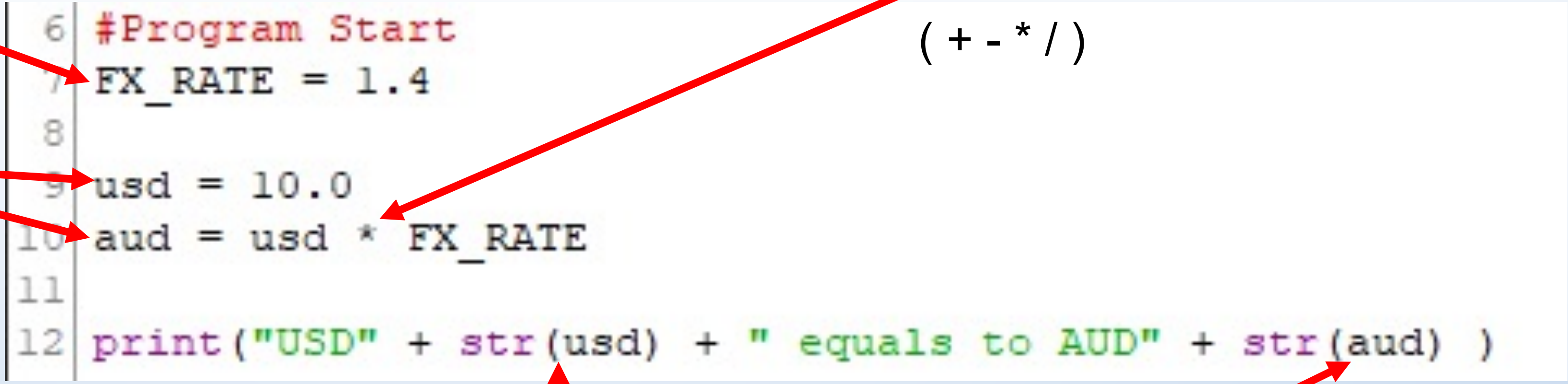
variables

Standard data types:

- Numbers
  int – 1,2,3,0,99,….
  float – 1.0, 3.1415, ….
- String
- List

operator

(+ - * /)



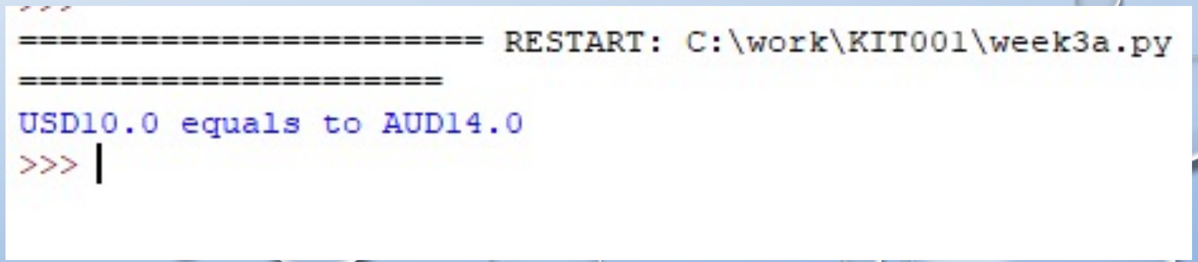
```python
#Program Start
FX_RATE = 1.4

usd = 10.0
aud = usd *  FX_RATE

print("USD" + str(usd) + " equals to AUD" + str(aud) )
```

Concatenate the string and print out variables.

Result:



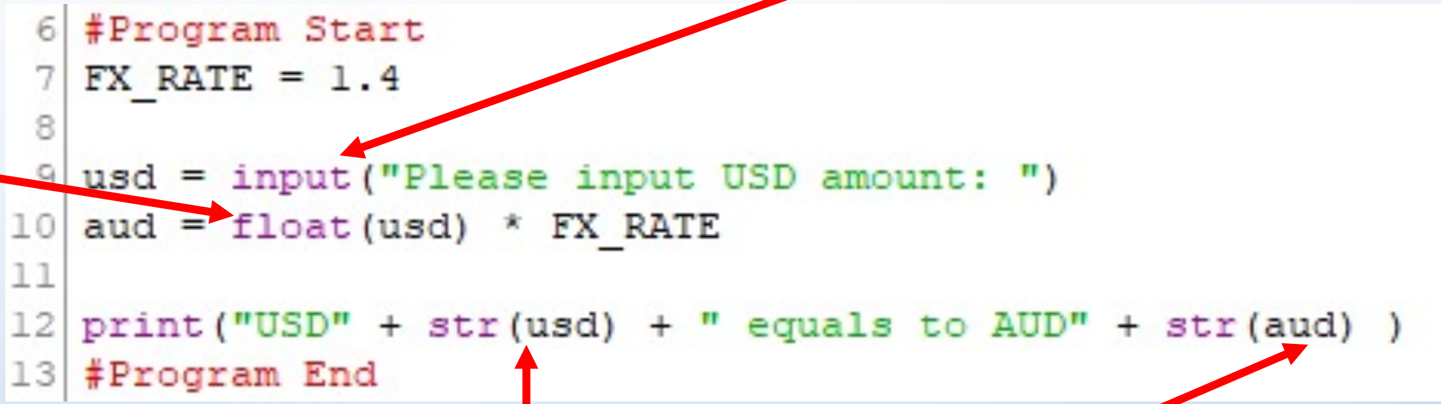
```
====================== RESTART: C:\work\KIT001\week3a.py
======================
USD10.0 equals to AUD14.0
>>> |
```

Python Syntax

input()
Ask for user input

float()
Change from **text** to **numeric** value

int()
Change from **text** to **integer**



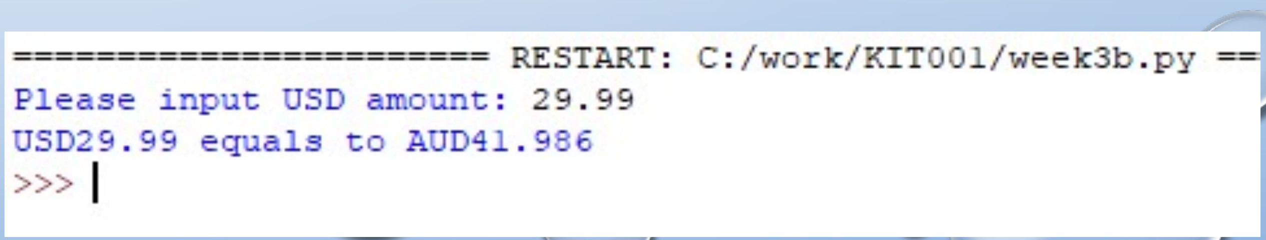
```
#Program Start
FX_RATE = 1.4

usd = input("Please input USD amount: ")
aud = float(usd) * FX_RATE

print("USD" + str(usd) + " equals to AUD" + str(aud) )
#Program End
```

Concatenate the string and print out variables.

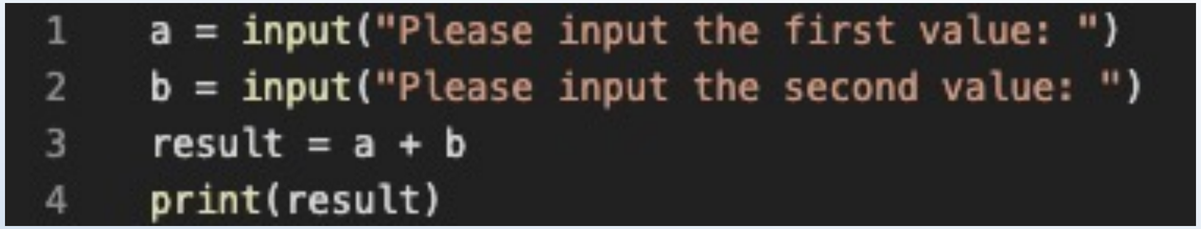
Result:



```
======================= RESTART: C:/work/KIT001/week3b.py ==
Please input USD amount: 29.99
USD29.99 equals to AUD41.986
>>> |
```

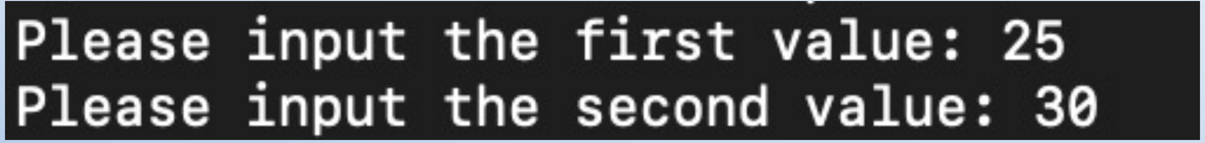
Let’s predict the output

You are given this program statements:



```
a = input("Please input the first value: ")
b = input("Please input the second value: ")
result = a + b
print(result)
```

User’s inputs are as below,



```
Please input the first value: 25
Please input the second value: 30
```

Result will be…?

A) 55

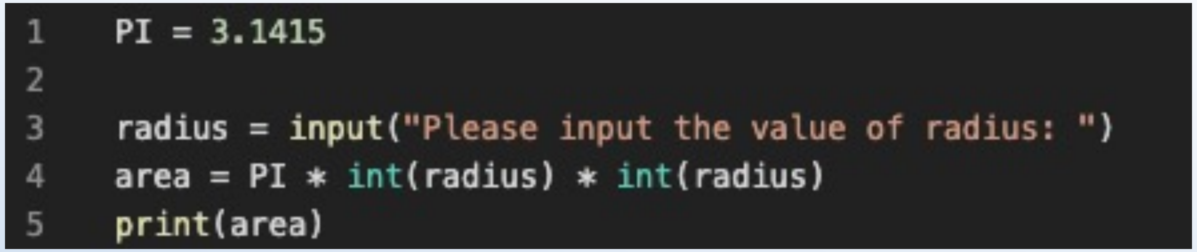
B) 2530

C) 3025

D) result

Let’s predict the output

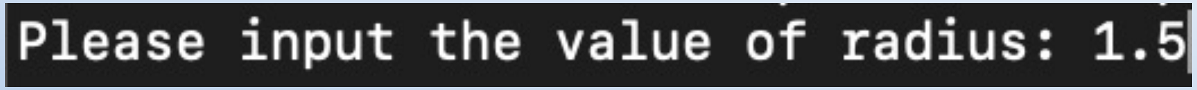
You are given this program statements:



```
PI = 3.1415

radius = input("Please input the value of radius: ")
area = PI * int(radius) * int(radius)
print(area)
```

User’s inputs are as below,



```
Please input the value of radius: 1.5
```

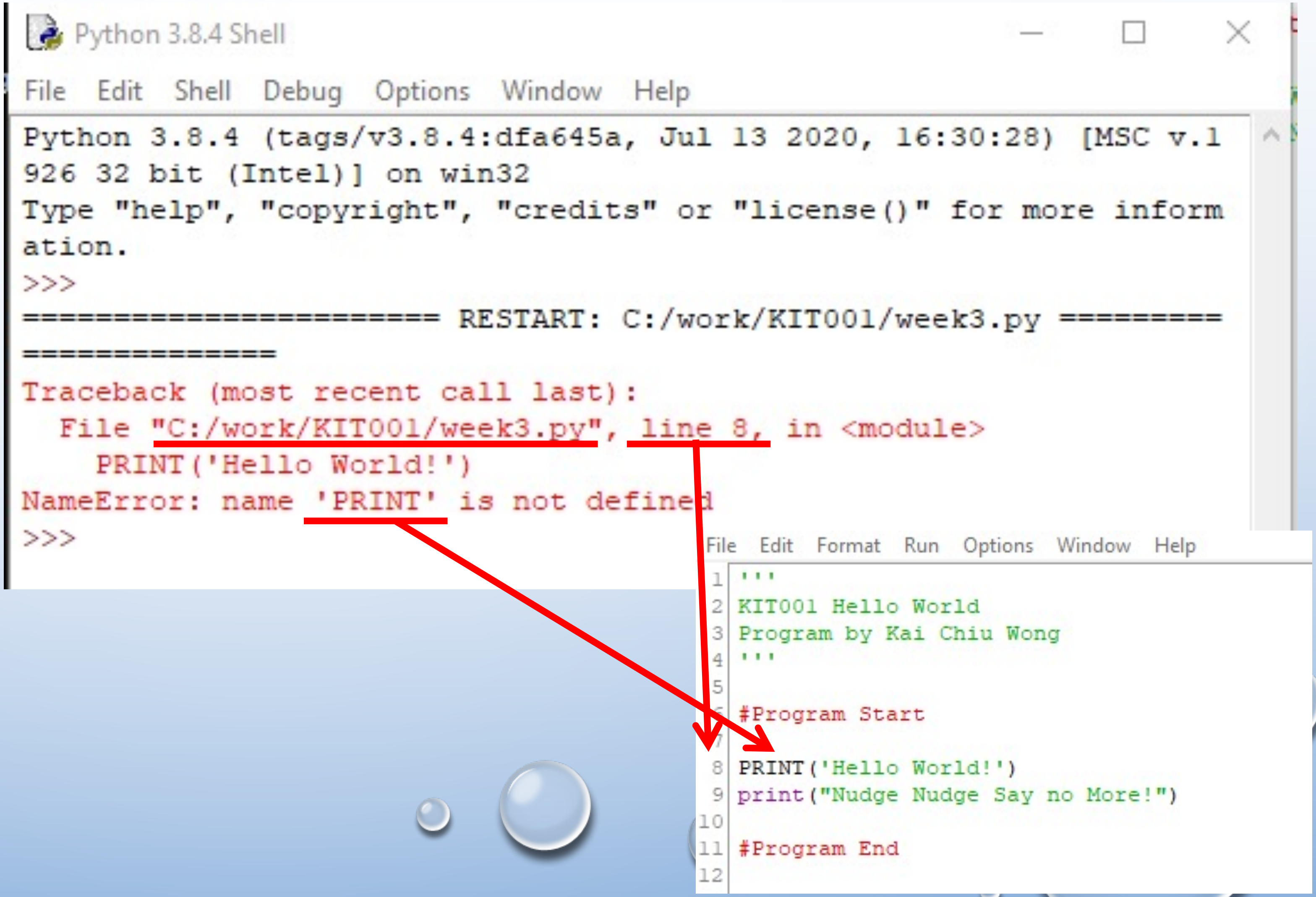
Result will be…?

```
A) 3.1415
B) 7.068375
C) 12.566
```

D) *Program Error*

Program Debug

- Errors are always here when doing development
- Make sure you can manage a skill to debug
  - Which file?
  - Which line?
  - Why?



Python 3.8.4 Shell
File Edit Shell Debug Options Window Help

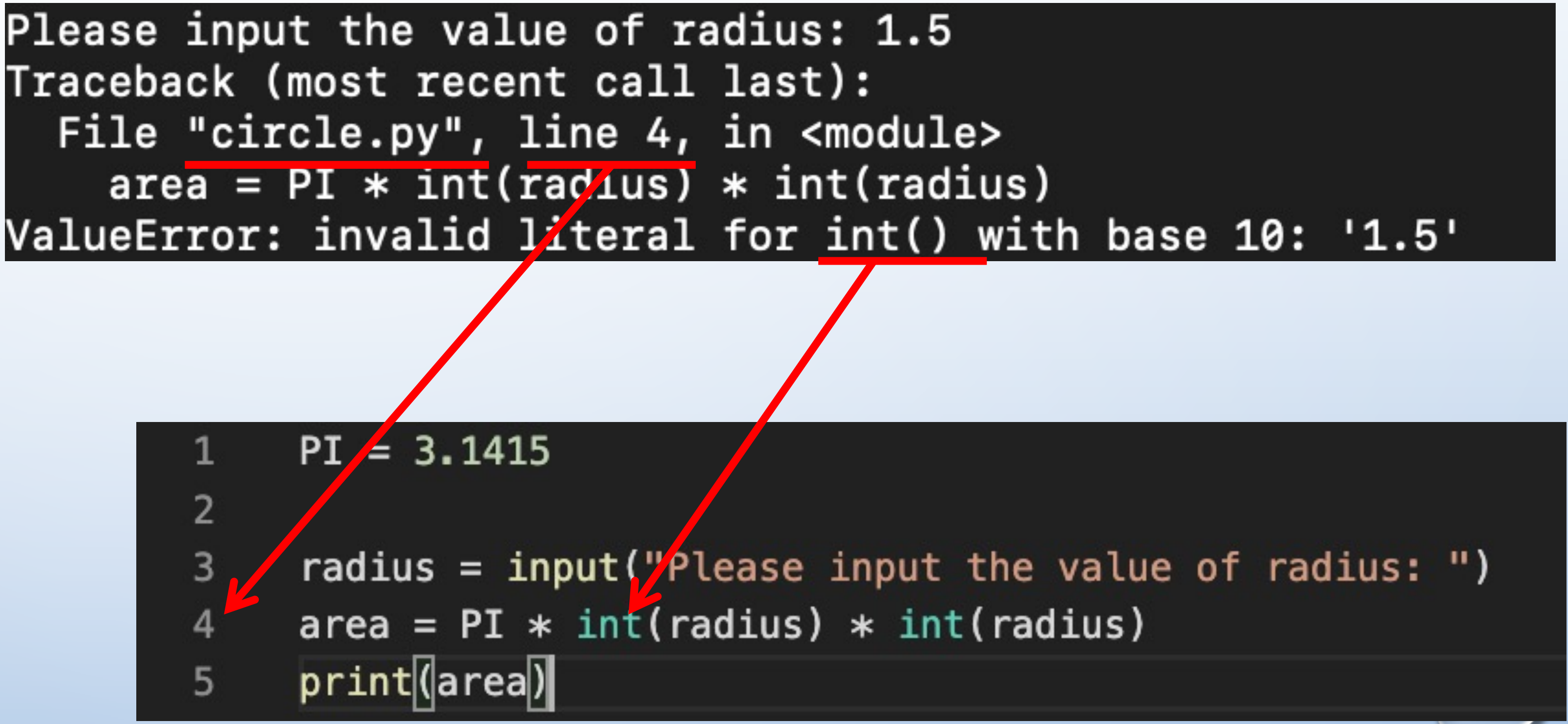
```
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:30:28) [MSC v.1
926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more inform
ation.
>>>
======================= RESTART: C:/work/KIT001/week3.py =========
==============
Traceback (most recent call last):
  File "C:/work/KIT001/week3.py", line 8, in <module>
    PRINT('Hello World!')
NameError: name 'PRINT' is not defined
>>>
```

File Edit Format Run Options Window Help

```
1  '''
2  KIT001 Hello World
3  Program by Kai Chiu Wong
4  '''
5
6  #Program Start
7
8  PRINT('Hello World!')
9  print("Nudge Nudge Say no More!")
10
11 #Program End
12
```

Program Debug

- Errors are always here when doing development
- Make sure you can manage a skill to debug
    - Which file?
    - Which line?
    - Why?



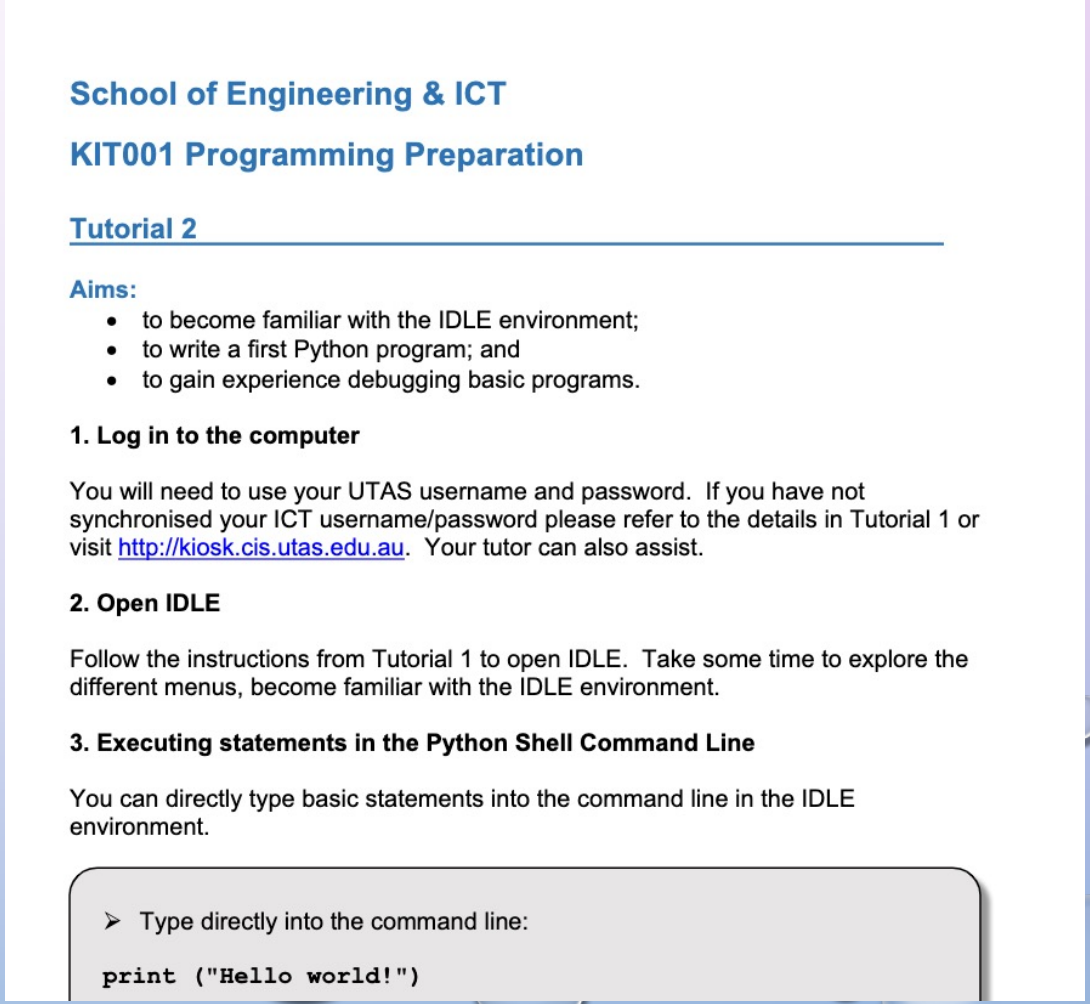
```
Please input the value of radius: 1.5
Traceback (most recent call last):
  File "circle.py", line 4, in <module>
    area = PI * int(radius) * int(radius)
ValueError: invalid literal for int() with base 10: '1.5'
```

```
1    PI = 3.1415
2
3    radius = input("Please input the value of radius: ")
4    area = PI * int(radius) * int(radius)
5    print(area)
```

Review and Practise

- Open **KIT100 Tutorial Two.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 2

Aims:

- to become familiar with the IDLE environment;
- to write a first Python program; and
- to gain experience debugging basic programs.

1. Log in to the computer

You will need to use your UTAS username and password. If you have not synchronised your ICT username/password please refer to the details in Tutorial 1 or visit http://kiosk.cis.utas.edu.au. Your tutor can also assist.

2. Open IDLE

Follow the instructions from Tutorial 1 to open IDLE. Take some time to explore the different menus, become familiar with the IDLE environment.

3. Executing statements in the Python Shell Command Line

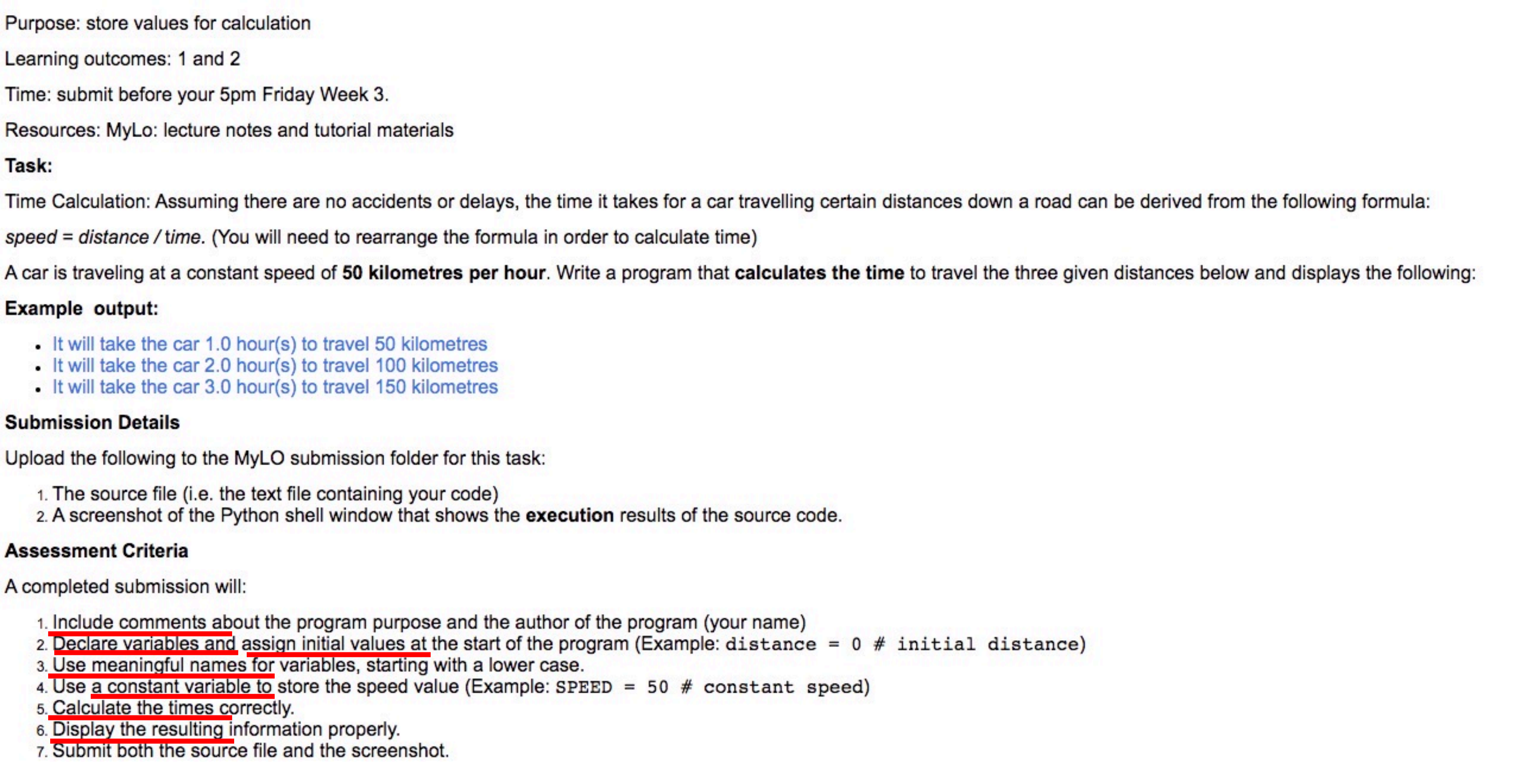
You can directly type basic statements into the command line in the IDLE environment.

➢ Type directly into the command line:

```
print ("Hello world!")
```

Portfolio Tasks

- **3.1PP Simple Calculation Expression**



Purpose: store values for calculation

Learning outcomes: 1 and 2

Time: submit before your 5pm Friday Week 3.

Resources: MyLo: lecture notes and tutorial materials

Task:

Time Calculation: Assuming there are no accidents or delays, the time it takes for a car travelling certain distances down a road can be derived from the following formula:

speed = distance / time. (You will need to rearrange the formula in order to calculate time)

A car is traveling at a constant speed of **50 kilometres per hour**. Write a program that **calculates the time** to travel the three given distances below and displays the following:

Example output:

- It will take the car 1.0 hour(s) to travel 50 kilometres
- It will take the car 2.0 hour(s) to travel 100 kilometres
- It will take the car 3.0 hour(s) to travel 150 kilometres

Submission Details

Upload the following to the MyLO submission folder for this task:

1. The source file (i.e. the text file containing your code)
2. A screenshot of the Python shell window that shows the **execution** results of the source code.

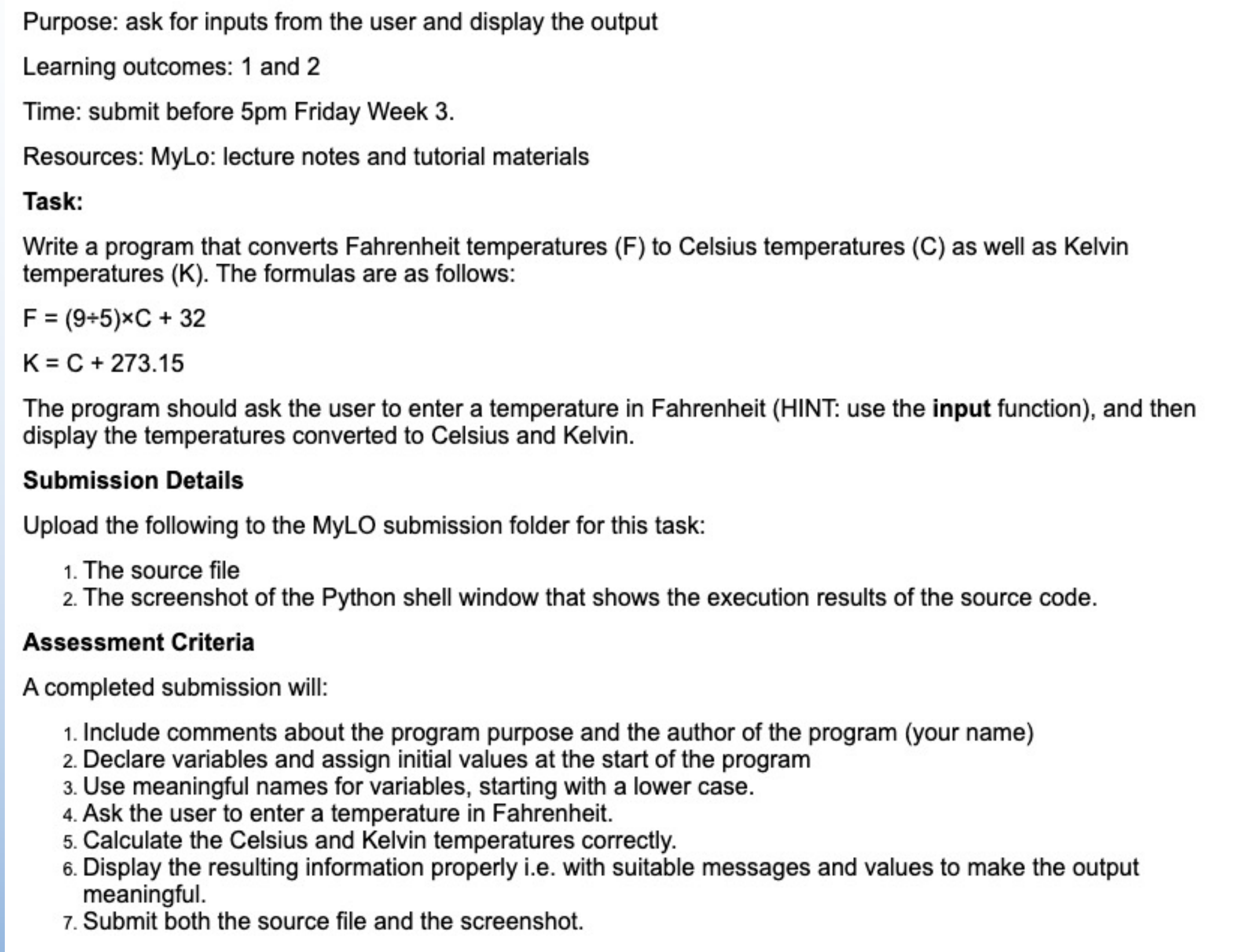
Assessment Criteria

A completed submission will:

1. Include comments about the program purpose and the author of the program (your name)
2. Declare variables and assign initial values at the start of the program (Example: `distance = 0 # initial distance`)
3. Use meaningful names for variables, starting with a lower case.
4. Use a constant variable to store the speed value (Example: `SPEED = 50 # constant speed`)
5. Calculate the times correctly.
6. Display the resulting information properly.
7. Submit both the source file and the screenshot.

Portfolio Tasks

- **3.2PP User Input**



Purpose: ask for inputs from the user and display the output

Learning outcomes: 1 and 2

Time: submit before 5pm Friday Week 3.

Resources: MyLo: lecture notes and tutorial materials

Task:

Write a program that converts Fahrenheit temperatures (F) to Celsius temperatures (C) as well as Kelvin temperatures (K). The formulas are as follows:

$F = (9 \div 5) \times C + 32$

$K = C + 273.15$

The program should ask the user to enter a temperature in Fahrenheit (HINT: use the **input** function), and then display the temperatures converted to Celsius and Kelvin.

Submission Details

Upload the following to the MyLO submission folder for this task:

1. The source file
2. The screenshot of the Python shell window that shows the execution results of the source code.

Assessment Criteria

A completed submission will:

1. Include comments about the program purpose and the author of the program (your name)
2. Declare variables and assign initial values at the start of the program
3. Use meaningful names for variables, starting with a lower case.
4. Ask the user to enter a temperature in Fahrenheit.
5. Calculate the Celsius and Kelvin temperatures correctly.
6. Display the resulting information properly i.e. with suitable messages and values to make the output meaningful.
7. Submit both the source file and the screenshot.