**Worksheet activity 2\_week 1**

**By the end of these exercises, you should be able to:**

* Create and use variables to store information in program
* Do simple maths calculation e.g. +, -, \*, /
* Understand the 3 basic variable types: string, integer and float
* Convert a variable's type to another (if it can be done, e.g. a string "3" to an integer 3).
* Write comments in program code
* Use string methods with string variables to create/modify new string

**New Terminologies:**

* Variable(s) – a storage location with a name. You can create variables to store information in computer program.
* Concatenation – Connecting strings together.

Programming Tip:  
Pay attention to detail when you read and type code. It will help you learn faster and save you a lot of time and errors

# Exercise 1—Print Statement.

Write a Python program to print greeting messages to attendees of a seminar. The program/script to print the following message **(please note that you have to use one print command only)**

Dear Seminar Participants,

Thank you for taking part in the upcoming seminar on

“Web Programming”.

The seminar will begin each day at 10:30 am with lunch and will end at 4 pm after evening tea each day.

Registration starts 10 am.

Best Wishes,

Seminar Organizer

# Exercise 2— Variables

Type the following code into a new Python program and run it. Then examine the code to see how it works.

salary = 53400

tax = salary \* 0.125

someString = "The amount of tax to pay: "

print(someString, tax)

interest = salary \* 0.04

print("Interest from saving my salary into a bank", interest)

**Question**

|  |  |
| --- | --- |
|  | **Your answer** |
| How many variables are there in this program? | **4 variables(salary, tax, somestring and interest)** |
| Type the code into a new program in Python IDLE and try it to see what it does.  Did it work the way you expected it? | **Yes the code worked as it was expected and as per my understanding when you are directly calling the variable some string and has defined it on top so no quotes is required to print a string,** |

**Read the Right Hand Side first**

Whenever you read a statement that has a '=' operator in it, read the right hand side (RHS) of the equal sign (=) first.

'=' is an assignment operator; that puts the results on the RHS into a variable (usually) on the LHS. To understand the first line of the program above, you read salary \* 0.125 first. The result of the calculation will be *assigned* to the variable tax on the LHS.

# Exercise 3 — Print Statement and Mathematical operation

Read the following example and try to identify the parts that does maths calculation.

Then copy and paste the following code into a new python program and run it to see what output it produces.

print("If a 2000 pound pregnant hippo gives birth to a 100 pound calf,")

print("but then eats 50 pounds of food, how much does she weigh?")

input("Press the enter key to find out.")

print("2000 - 100 + 50 =", 2000 - 100 + 50)

print("\nIf an adventurer returns from a successful quest and buys each of")

print("6 companions 3 bottles of ale, how many bottles are purchased?")

input("Press the enter key to find out.")

print("6 \* 3 =", 6 \* 3)

print("\nIf a restaurant check comes to 19 dollars with tip, and you and")

print("your friends split it evenly 4 ways, how much do you each throw in?")

input("Press the enter key to find out.")

print("19 / 4 =", 19 / 4)

print("\nIf a group of 4 pirates finds a chest full of 107 gold coins, and")

print("they divide the booty evenly, how many whole coins does each get?")

input("Press the enter key to find out.")

print("107 // 4 =", 107 // 4)

# Exercise 4 — Mathematics Operators

To do maths in Python, we can use the following Maths operators! (Don’t they look like the ones you actually use in Maths?)

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Description | Example | Evaluates To |
| + | Addition | 7 + 3 | 10 |
| - | Subtraction | 7 - 3 | 4 |
| \* | Multiplication | 7 \* 3 | 21 |
| / | Division (True) | 7 / 3 | 2.3333333333335 |
| // | Division (Integer) | 7 // 3 | 2 |
| % | Modulus | 7 % 3 | 1 |
| \*\* | ?? | 7\*\*3 | 343 |

Table 1 – Maths operators

**Questions**

|  |  |
| --- | --- |
|  | **Your answer** |
| What does \*\* do? |  |
| Use Python code to find the result of the following calculations |  |
| 21 \* (366+41) | **8547** |
| 33 \* 4 \* 2.5 \* 6 | **1980.0** |
| 1 + 29 + 33 + 40 \* (11+2) | **583** |
| The remainder of 10/4 | **2.5** |
| 4/2/3 | **0.6666666666666666** |

# Exercise 5—Input and Processing statements

Can you extend the following program to ask for user's age and occupation and then display them?

1. # Greeter
2. name = input("Hi, what is your name?")
3. # Add few lines to ask for user's age and occupation
4. print("Hi,", name)
5. # Add few lines to display user's age and occupation
6. input("\n\nPress the enter key to exit.")

**Note:**

* Remember to read the RHS of the '=' operator first and then the LHS.
* The input() function prints one string argument as a prompt and waits for user to enter something on the keyboard. Once user typed something and presses the Enter key, the information will be assigned/stored into the variable on the LHS.
* input() function produces strings. If you need the input as an integer so you can use it as a number, you need to convert its type first: string to integer.

# Exercise 6— Input and Processing statements

Write a Python program to find the average of two given numbers (sum of two numbers divided by two).

Sample screen (**200 and 100 are example inputs, your program should work with any integer numbers**)

What is the first number? **200**

What is the second number? **100**

The average of two numbers is equal to 150.0

# Exercise 7— Variable's type

Type the following code into a new Python program and run it to see how it works

number = input("Enter a number:")

print(number)

result = number \* 3;

print(result)

|  |  |
| --- | --- |
|  | Your answer |
| What do you think the program will print out if 3 is entered? | It will repeat the same number 3 times |
| Was the result different than what you expected? What do you think is the reason behind? | It is because we did not mention the integer as in result? |

Sometimes a program doesn’t do what we want it to do when the types of the variables are not right!

Add the following line to after calling the input function in the program (insert it after Line 1) and see how it works.

number = int(number)

Alternatively, you can change Line 1 to:

number = int(input("Enter a number:"))

In either of these two lines, we make sure we convert the entered value to integer first and then store the value to the number variable.

# Exercise 8—Triple Quoted Strings

age = input("What is your age")

age = int(age)

age = age + 1

print ("You will be", age, "next year")

futureAge = age + 9

print("You will be", futureAge, "in 10 years")

1. What does the statement **age = age + 1** do?

It is adding another year with the current age.

1. As a shortcut, **age = age + 1** can also be written as **age+=1**

# Exercise 9 – What is the type of the following variables?

a = "2"

b = 4.3

c = 6

d = 'abc'

When computer stores data, it stores them in different types (float, integer, string and etc…) Given the code above, what do you think is the type of the a,b,c,d variables?

|  |  |
| --- | --- |
|  | Your answer |
| a | String |
| b | float |
| c | integer |
| d | string |

The following functions can be used to convert values between types!

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Description | Example | Returns |
| float(*x*) | Returns a floating-point value by converting ***x*** | float("10.0") | 10.0 |
| int(*x*) | Returns an integer value by converting ***x*** | int("10") | 10 |
| str(*x*) | Returns a string value by converting ***x*** | str(10) | '10' |

# Exercise 10 – Float variables

Write a new Python program to ask its user his/her weekly rent, bus fare and grocery costs. Then calculate and print the total by adding these 3 costs up.

Your program should allow inputs with decimal places!

Example (120.55, 40.35 and 20.99 are example inputs):

How much is your rent (weekly)? **120.55**

How much do you pay for bus fare (weekly)? **40.35**

How much did you spend on grocery last week? **20.99**

Your total weekly cost: 181.89

# Exercise 11 – Tipper program

Write a new Python program (Tipper) where user enters a restaurant bill total. The program should then display two amounts: a 15 percent tip and a 20 percent tip.

Example (120 is an example input)

Please enter bill amount: **120**

15 percent tip: 18.0

20 percent tip: 24.0

Note: 15% is equivalent to 0.15 or

# Exercise 12 – Combining Food Names

Write a new program that allows user to enter his or her two favourite foods. The program should then print out a message as follows by joining strings together.

Example (pizza and cola are example input)

Please enter your favourite food: **pizza**

Please enter another favourite food: **cola**

A new food was created: pizzacola

# Exercise 13 – Car Salesman

Write a Car Salesman program where the user enters the base price of a car. The program should add on a bunch of extra fees such as tax, license, dealer prep, and destination charge. Make tax and license a percent of the base price. The other fees should be set values. Display the actual price of the car once all the extras are applied.

Example:

What is the car model: **TOYATA**

What is the year of the car: **2015**

Please enter the base price of the car: **10000**

Tax (12.5%) for the car is 1250.00

License fee (fixed) is 100.00

Dealer prep (fixed) is 500.00

Destination charge (fixed) is 300.00

The total price of TOYATA 2015 (incl. all fees): 12150.00

Submit this to Moodle when you finish it.

But before submitting, put comments into your code. Use **# (hash)** key to put comment in code.

Use the following template. Modify it to include your own name and other info.

# Author name: Tony Kuo

# Date: 21 July 2014

# Quick Description: A birthday program

Comment is for yourself and others who read your code. Good comments help yourself (author) and other programmers understand your code better. From now one, put into practice to write some comments in your code whenever you think the code is not easy to read or is not really self-explanatory!