Introduction to Programming and Problem Solving

Session 2: Beginning

**Programming with Python**

**INTENDED**

**LEARNING OUTCOMES**

**By the end of this week you will be able to…**

* **FLOWCHARTS**

Sketch flowcharts for basic computer programs

* **CODE**

Write Python code for basic computer programs that use the following:

* Read user input
* Write to standard output
* Perform basic calculations
* Use constants
* Use variables
* Use different data types
* Convert between data types
* Use operators

**QUICK OVERVIEW**

This workbook introduces students to the basics of programming using Python. Students will learn how to produce simple python programs using IDLE (a basic code editor). This workbook provides students with the opportunity to complete the following activities:

1. Printing with Python **(~20 minutes)**

2. Reading with Python **(~10 minutes)**

3. Data Types and Operators in Python **(~60 minutes)**

4. Preparing for next week **(~120 minutes)**

**INTRODUCTION**

Python is a popular and widely used programming language. It is classified as a high-level language which means that it closely resembles natural (human) languages. This is in contrast to low-level languages which more closely resemble the computer’s architecture and so are more difficult for humans to understand.

Python has been created with code readability in mind and so provides many constructs that help developers quickly and easily develop programs that are scalable and easy to understand.

There are many reasons for learning python including the following:

* Python is FUN to learn!
* It has a lower barrier to learning than many other languages which makes it quicker and easier to learn
* Python is a modern, high level language that makes code comprehension easier than a number of other languages
* Python is generally portable across different operating systems
* Python comes with many libraries to help you get going
* Python is free, well documented and has a well-established active community

Maslow’s Hammer:

“*If the only tool you have is a hammer,   
every problem looks like a nail*”

Finally, learning a new programming language allows us to look at problems in different ways and discover more suitable solutions.

**INTRODUCTION**

What is python?

Why learn python?

**ACTIVITY 1: PRINTING WITH PYTHON**

We will begin by learning how to create programs that can “communicate” with the user by printing characters on the screen. We will use python’s built-in function print to print characters on the screen. For now we do not need to concern ourselves with how print works but simply that it has been provided to us for the purpose of printing values.

**TASK 1.1: HELLO WORLD!**

Let us consider a simple example where we wish to print the message “Hello World!” to the screen.

To do this first launch IDLE (the code editor that comes with python). You should see an IDLE window similar to that below:

Now type the following into IDLE:

print(“Hello World!”)

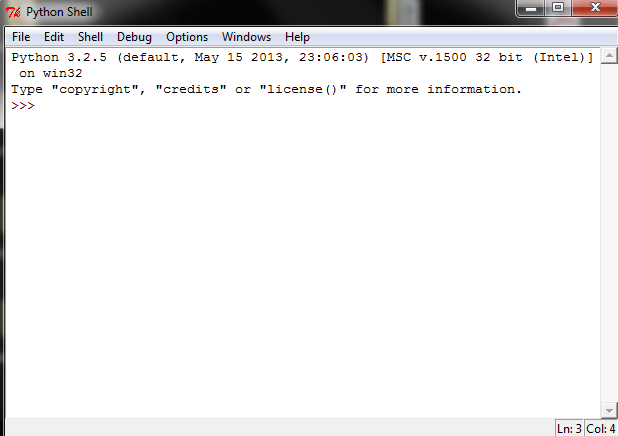
Press enter.

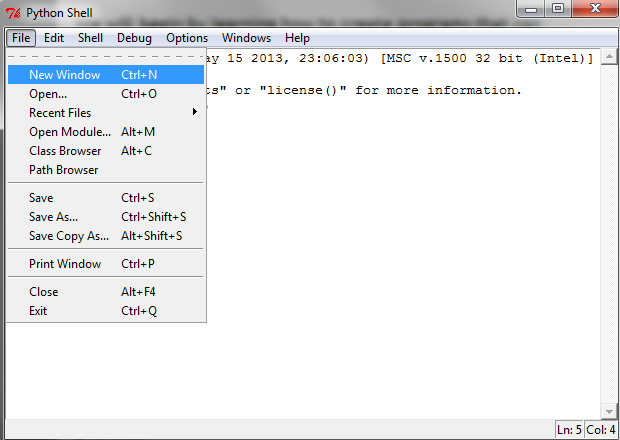
You will see that IDLE prints Hello World! on the next line

**ACTIVITY 1: PRINTING WITH PYTHON**

**(~20 minutes)**

Task 1.1: Hello World!

****

****

**ACTIVITY 1: PRINTING WITH PYTHON**

**- CONTINUED**

**TASK 1.2: THE ANSWER TO EVERYTHING**

Let us try another example.

We will print the message “The answer to the ultimate question of life, the universe and everything is 42” to the screen.

But this time we will create a new python file and save our work.

Start by clicking ‘File’ -> ‘New Window’ (or pressing Ctrl + n on the keyboard).

In the resulting window type:

print(“The answer to the ultimate question of life, the universe and everything is 42”)

Then save the file as answer.py in your “My Work” folder.

Finally, run your code by clicking on ‘Run’ -> ‘Run Module’ or by pressing ‘F5’ on your keyboard.

The message should be printed in your IDLE screen.

**ACTIVITY 1: PRINTING WITH PYTHON - CONTINUED**

Task 1.2: The Answer to Everything

**ACTIVITY 1: PRINTING WITH PYTHON**

**- CONTINUED**

**TASK 1.3: ESCAPE CHARACTERS AND SEPARATORS**

Try each of the following and note down what happens and why?

print(“The secret of getting ahead is”, “getting started”)

print(“There are”, 2 \*\* 32, “possibilities”)

print(“There are”, 2 \*\* 32, “possibilities”, sep=”- - -“)

print(“Start where you are.\nUse what you have.\nDo what you can.”)

print(“In order to succeed, \twe must first believe we can.”)

**Splitting the text on two parts, the output is not changing**

**Making a big space between output text**

**Making otput from the next line**

**Making the output with the result of calculation**

**Adding a separation between numbers and text**

**ACTIVITY 1: PRINTING WITH PYTHON - CONTINUED**

Task 1.3: Escape Characters and Separators

**ACTIVITY 1: PRINTING WITH PYTHON**

**- CONTINUED**

**TASK 1.4: A BOX**

We can print many lines of characters using multiple calls to print.

Let us assume that we wish to print a ‘box’ similar to that shown below:

What lines of code do we need to achieve the above?

**TASK 1.5: A CARTOON**

Let us now assume we wish to be a bit more creative and that we wish to print the cartoon face shown in the image on the next page:

**ACTIVITY 1: PRINTING WITH PYTHON - CONTINUED**

Task 1.4: A Box

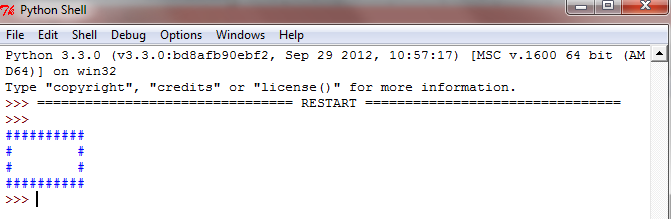
Task 1.5: A Cartoon

**print("0 0\n 88 \n000000000")**

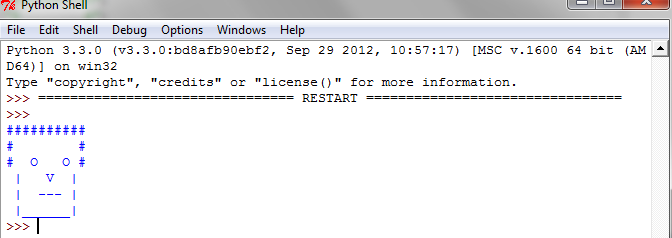
0 0

88

000000000

****

**print("#########\n# ^ ^ #\n# $ $ #\n# U # \n#\_\_\_\_\_\_\_# \n#########")**

****

**ACTIVITY 1: PRINTING WITH PYTHON**

**- CONTINUED**

What lines of code do we need to achieve the above?

**ACTIVITY 2: READING WITH PYTHON**

Python allows us to read user input using the built-in function input. We can assign the result of this built-in function to a variable.

A variable is simply a memory location that is reserved by your program for storing a value.

By assigning the value returned by input to a variable we make it available for use later in the code as demonstrated by the next task.

**ACTIVITY 1: PRINTING WITH PYTHON - CONTINUED**

**ACTIVITY 2: READING WITH PYTHON**

**(~20 minutes)**

**Int for integer, complete numbers**

**Float for decimal numbers**

**Str for string**

Here are some examples of using input.

1. Read in a whole number

userNumber = int(input ())

1. Read in a floating point number

userFloat = float(input ())

1. Read in a string

userString = str(input ())

Try each of the above with different values as input and print out the results. Note any observations below.

**ACTIVITY 2: READING WITH PYTHON**

**(~20 minutes)**

**print("What is your name?")**

**uStr = str(input())**

**print("Hello, ",uStr)**



**ACTIVITY 2: READING WITH PYTHON - CONTINUED**

Task 2.1: What is your name?

**ACTIVITY 2: READING WITH PYTHON**

**- CONTINUED**

**TASK 2.1: WHAT IS YOUR NAME?**

Draw a flowchart for a computer program that prompts the user with the question “What is your name?” and then uses the response to print “Hello ” followed by the person’s name.

Build this program using Python and provide your code below:



****

**ACTIVITY 2: READING WITH PYTHON**

**- CONTINUED**

**TASK 2.2: CARTOON EYES**

Design and code a program using Python which prompts the user to enter a character and then prints a cartoon face similar to that in activity 1 but where the eyes are replaced with the character supplied by the user. For example, if the user enters ‘X’, the cartoon face will look as follows:

Provide your flowchart below:

**ACTIVITY 2: READING WITH PYTHON – CONTINUED**

Task 2.2: Cartoon Eyes

**ACTIVITY 2: READING WITH PYTHON – CONTINUED**

Task 2.2: Cartoon Eyes - continued

**ACTIVITY 2: READING WITH PYTHON**

**- CONTINUED**

**TASK 2.3: CARTOON EYES**

Provide your code below.

**uStr = str(0)**

**print("type a symbol to put it in the eyes of the face shown below:")**

**print("#########\n# ^ ^ #\n#",uStr," ",uStr,"#\n# U # \n# ----- # \n#########")**

**uStr = str(input ())**

**print("#########\n# ^ ^ #\n#",uStr," ",uStr,"#\n# U # \n# ----- # \n#########")**

**Data type helps compiler understand what the data is for.**

**ACTIVITY 3: DATA TYPES AND OPERATORS**

**(~60 minutes)**

Task 3.1: Data types

**ACTIVITY 3: DATA TYPES AND OPERATORS IN PYTHON**

**TASK 3.1: Data types**

Your teacher will discuss data types and operators. After your teacher has finished this discussion, answer the following questions:

What is a data type? What data types available in python and how are they used?

**Operators are to calculate, compare data types**

**ACTIVITY 3: DATA TYPES AND OPERATORS IN PYTHON**

**TASK 3.2: Operators**

What is an operator? What operators are available in python and how are they used? Include an example use of each type of operator.

**ACTIVITY 3: DATA TYPES AND OPERATORS**

**(~60 minutes)**



**ACTIVITY 3: DATA TYPES AND OPERATORS IN PYTHON**

**TASK 3.3: BMI**

Your final task is to design and create a program that can calculate a person’s body mass index (BMI). Your program should function in a similar manner to the following example:

<http://www.nhs.uk/chq/Pages/how-can-i-work-out-my-bmi.aspx?CategoryID=51>

Design the program using a flowchart:

**ACTIVITY 3: DATA TYPES AND OPERATORS**

**(~60 minutes)**

**print("Type your weight in kg:")**

**kg = float(input())**

**print("Type your height in metres:")**

**met = float(input())**

**print("your BMI is:",round(kg / met / met,2))**

**if ((kg / met / met) < 18) : print ("u are underweight")**

**elif ((kg / met / met) <25) : print ("u are ok")**

**else : print ("u are fat")**

**ACTIVITY 3: DATA TYPES AND OPERATORS IN PYTHON - CONTINUED**

**TASK 3.3: BMI**

Provide the code for you BMI calculator below:

**ACTIVITY 3: DATA TYPES AND OPERATORS**

**(~60 minutes)**

**SUMMARY**

This week we have been introduced to the basics of programming in python using IDLE. We have learnt how we can write a program to communicate with the user.

**PREPARING FOR THE NEXT SESSION**

Before the start of the next session you should attempt to design (using flowchart) and code (in python) a program for each of the following:

1. *A program that reads in a number and then calculate the following formula:  
    2n + 1  
   where n is the number entered by the user*
2. *A program that reads in the base and height of a triangle and calculates the area of the triangle*

Upload your solutions on to SOL using the submission link when done.

A short formative assessment will take place next week. You should review all the work covered so far in preparation for the formative assessment.

**SUMMARY**