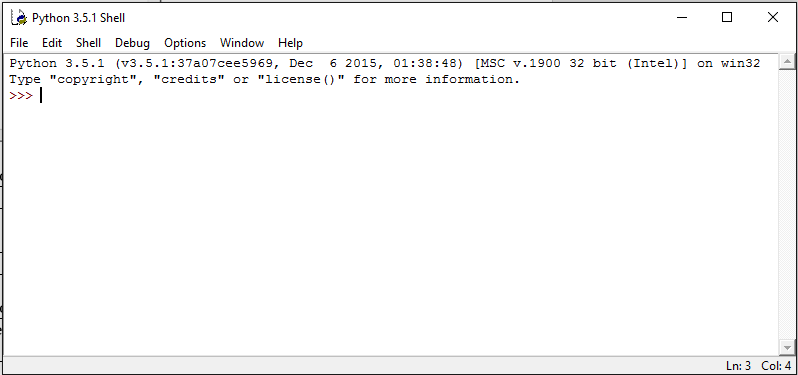
Name: ……………………………………………….. ( ) Class: ……… Date: …………………..

|  |  |  |
| --- | --- | --- |
| **4.1** | **Program Development** | **Python Syntax and Variables** |

When you run Python IDLE, the following screen will be shown. This is known as the **Python shell**. From it, some useful information is displayed. The first line tells you about the Python version that you have installed. At the bottom of the screen, the cursor’s position is displayed in line and column format. This information will be useful for locating a particular line when fixing errors.



**Python version information**

**Line and column number of cursor**

**Hello World**

1. Type the following at the Python shell (without the >>>) and press Enter:

|  |
| --- |
| >>> **print("Hello, World")** |

Write down the output that appears in the space below:

|  |
| --- |
|  |

1. Type the following at the Python shell and press Enter. You will be asked to enter your name. Do so and press Enter again:

|  |
| --- |
| >>> **name = input("What is your name? ")** |

After entering your name, type the following at the Python shell and press Enter:

|  |
| --- |
| >>> **print("Hello, " + name)** |

Write down the output that appears in the space below:

|  |
| --- |
|  |

**Creating and saving a Python file**

You can also create and save a Python file to run the above commands.

1. On the Python shell, click File, then New File. You will notice a text editor appears. Type the following two lines into the text editor:

|  |
| --- |
| **name = input("What is your name? ")**  **print("Hello, " + name)** |

To run the file, press F5. You will be prompted to save the file. Choose a suitable location to save the file. By default, the Python file is saved with the extension .py. Notice that the output is now displayed at the Python shell.

**Indentation**

Python programs are structured through indentation. This means that code blocks in Python are defined by their indentation. If multiple lines of code are indented by the same number of spaces, they belong to the same block. This makes code more readable and easier to understand.

Note that Python does not allow mixing the use of tabs and spaces for indentation.

1. What is the recommended number of spaces per level of indentation? ……………
2. What is wrong with lines 2 and 3 of the following code entered in the Python shell? (Note that the characters are shown in a grid for clarity.)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| > | > | > | a | = | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| > | > | > |  | b | = | 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| > | > | > |  | p | r | i | n | t | ( | a | , | b | ) |  |  |  |  |  |  |  |  |

………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………

1. Write down the corrected code below:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| > | > | > | a | = | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| > | > | > |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| > | > | > |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Create a Python file and enter the following code into the file. Save the file as testprint.py and run it.

|  |
| --- |
| **if True:**  **print("Line 1")**  **print("Line 2")**  **print("Line 3")**  **print("Line 4")**  **print("Line 5")** |

What is wrong with the code?

……………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………

1. Write down the corrected code below:

|  |
| --- |
|  |

1. The following code is not indented properly and parentheses are missing.

|  |
| --- |
| **if True:**  **print "True"**  **else:**  **print "False"** |

Write down the corrected code below:

|  |
| --- |
|  |

1. Enter and run the following code:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | **BMI = 22.0**  **if BMI > 18.5:**  **if BMI > 23:**  **if BMI > 27.5:**  **print("Overweight")**  **else:**  **print("Moderately Overweight")**  **else:**  **print("Healthy Weight")**  **else:**  **print("Underweight")** |

*(Line numbers are provided for reference only and should not be entered.)*

1. Count the number of spaces used as indentation in:
   1. Line 3 ……………
   2. Line 5 ……………
   3. Line 9 ……………
2. What do you think the program does?

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

**Comments**

A hash sign (#) that is not inside a string literal begins a comment. The Python interpreter disregards comments and does not print them. Comments are useful for documenting program code.

Predict what will be printed by the following code, then try it out and record the actual results:

|  |
| --- |
| **# This is a comment**  **print("This is not a comment.")** |

1. Expected Output:

|  |
| --- |
|  |

1. Actual Output:

|  |
| --- |
|  |

An inline comment is a comment on the same line as a statement. Inline comments should be separated by at least two spaces from the statement. They should also start with a # and a single space. Inline comments may be confusing so use them sparingly.

For example:

|  |
| --- |
| **word = "Wonderful!" # This is an inline comment** |

You can comment multiple lines as follows:

|  |
| --- |
| **# This is a comment.**  **# This is a comment too.**  **# This is a comment again.** |

However, for comments that span multiple lines, it may be easier to create them as documentation strings that are surrounded by """:

|  |
| --- |
| **"""This whole chunk is a documentation string.**  **This is a comment.**  **This is a comment too.**  **This is a comment again.**  **"""** |

Predict what will be printed by the following code, then try it out and record the actual results:

|  |
| --- |
| **#This is a comment**  **#This is also a comment. It won't print(anything)**  **print("This is not a comment.") #although this is...**  **"""**  **These**  **are**  **all**  **comments**  **in a documentation string**  **"""**  **print("But the comments are finished now.")** |

1. Expected Output:

|  |
| --- |
|  |

1. Actual Output:

|  |
| --- |
|  |

**Rules for Python Identifiers**

A Python identifier is a name used to identify a variable, function, module or other objects.

1. When naming an identifier, the following rules must be observed:

* Begin with a:
  + lowercase character …………………………… or
  + uppercase character …………………………… or
  + …………………………… and
* Followed by …………………………… characters or '\_'
* Do not use ……………………………………………………words
* Python is ……………………………………………………………

**Variables**

Variables are locations that we can use to store data in.

1. Which of these variable names are allowed in Python? Put a tick for names that are allowed and a cross for names that are not allowed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| box |  |  | postal code |  |  | pi |  |
| Box |  |  | postal\_code |  |  | PI |  |
| BOX |  |  | postal.code |  |  | alex |  |
| Box2 |  |  | postalCode |  |  | Alex |  |
| 1Box |  |  | postal# |  |  | 3Alex |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GreatBigVariable |  |  | 6x |  |
| greatBigVariable |  |  | x6x |  |
| great\_big\_variable |  |  | all% |  |
| great.big.variable |  |  | #left |  |

To assign a value to a variable, use the = sign.

For example, num = 10 means that the value of 10 is assigned to a variable named num.

Predict what will be printed by the following code, then try it out and record the actual results:

|  |
| --- |
| >>> **apples = 2**  >>> **oranges = 7**  >>> **total\_fruits = apples + oranges**  >>> **print(total\_fruits)** |

|  |
| --- |
|  |

1. Expected Output:

|  |
| --- |
|  |

1. Actual Output:

|  |
| --- |
|  |

1. Predict what will be printed by the following code, then try it out and record the actual results. The code for the first row is entered before the code in the second row, and so on:

|  |  |  |
| --- | --- | --- |
| **Code** | **Expected Output** | **Actual Output** |
| >>> **x = 8**  >>> **print(x)** |  |  |
| >>> **x = x + 1**  >>> **print(x)** |  |  |
| >>> **x = x + x**  >>> **print(x)** |  |  |

**Constants**

A constant refers to a variable whose value cannot be changed. To define a constant in Python, the convention is to use capital letters with underscores separating words. For example, MAX\_SCORE and TOTAL are constants.

1. Which of these names are likely to be for constants? Put a tick for names that are likely to be for constants and a cross for names that are not.

|  |  |
| --- | --- |
| Area |  |
| AREA |  |
| Sum\_of\_triangles |  |
| SUM\_OF\_TRIANGLES |  |

1. Explain what is wrong in the following code:

|  |
| --- |
| **print(a)**  **a = 45** |

……………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………

1. Explain what is wrong in the following code:

|  |
| --- |
| **x = 4**  **y = 5**  **a = 3(x + y)**  **print(a)** |

……………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………