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

|  |  |  |
| --- | --- | --- |
| **4.1** | **Program Development** | **Numeric Data Types** |

Python supports two numerical types: integers and floating-point numbers.

|  |  |  |
| --- | --- | --- |
| **Numerical type** | integer (int) | floating-point (float) |
| **Example** | 10 | 2.56 |

**Entering integers and floating-point numbers**

1. To enter an integer directly in Python, type out its digits. For instance, the following code assigns the integer value 5 to a variable named a:

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

To enter a floating-point number directly in Python, type out its digits including its decimal point. For instance, the following code assigns the floating-point value 6.0 to a variable named b:

|  |
| --- |
| >>> **b = 6.0** |

To find out the type of a variable or value, use the type() function:

|  |
| --- |
| >>> **type(a)**  >>> **type(b)** |

Record the output of type(a) and type(b) here:

|  |
| --- |
|  |

1. There are other ways to enter ints or floats. For instance, floating-point numbers can also be entered in the form AeB, which represents A × 10B.

Classify the following values as either ints or floats. You may enter the values in Python IDLE and use the type() function to check your answers.

|  |  |  |
| --- | --- | --- |
| **No** | **Example** | **Numeric Type (int or float)** |
| a) | 10 |  |
| b) | 0.02 |  |
| c) | 99 |  |
| d) | 15.30 |  |
| e) | -786 |  |
| f) | -21.9 |  |
| g) | 2e15 |  |
| h) | -80. |  |
| i) | 33E-33 |  |
| j) | 435600 |  |

**Adding an integer and a floating-point number together**

We can perform arithmetic operations that mix integer and floating-point numbers together. For instance, predict what will be printed for the following code, then write down the actual output:

|  |
| --- |
| >>> **a = 5**  >>> **b = 6.0**  >>> **a + b** |

1. Expected Output of a + b:

|  |
| --- |
|  |

1. Actual Output of a + b:

|  |
| --- |
|  |

1. What is the resulting data type when you add an integer and a floating-point number together?

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

**Conversion between data types**

Python can change a floating-point number into an integer using the int() function. Note that this does not round the number up, it just takes the whole number part of the floating-point number.

The technical term for changing the type of data you are working with is called **type casting**.

1. For instance, the following code converts the floating-point number 4.2 to an integer and stores it in a variable c. Enter the code and write down the actual output:

|  |
| --- |
| >>> **c = int(4.2)**  >>> **c** |

Actual output:

|  |
| --- |
|  |

Likewise, Python can change an integer into a floating-point number using the float() function.

1. For instance, the following code converts the integer 8 to a floating-point number and stores it in a variable d. Enter the code and write down the actual output:

|  |
| --- |
| >>> **d = float(8)**  >>> **d** |

Actual Output:

|  |
| --- |
|  |

**Automatic conversion to float for division**

1. The example below demonstrates that Python automatically converts the result of dividing two integers to a floating-point number. Enter the following code and write down the actual output:

|  |
| --- |
| >>> **e = 22**  >>> **f = 7**  >>> **g = e / f**  >>> **g** |

Actual Output:

|  |
| --- |
|  |

1. The result of the previous calculation is a floating point number as dividing two whole numbers with each other is not guaranteed to result in a whole number. To force the result to be rounded down to a whole number, use the floor division operator // instead.

For instance, enter the following code and write down the actual output:

|  |
| --- |
| >>> **e = 22**  >>> **f = 7**  >>> **g = e // f**  >>> **g** |

Actual output:

|  |
| --- |
|  |

1. Explain how the following line of code can be simplified and why.

pi = float(3.14159)

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