Lesson 5

Input and Decision-Making

1. **KeyBoard Input**

Input in Python is very simple. Depending what you wish to input, you would use one of the following lines:

inputString = input()

and/or

aDecimal = float(input())

and/or

anInteger = int(input())

Note that you cannot input a decimal number and directly convert it to an integer (why would you want to?).

1. **If Statements**
2. The basic if statement has the following structure:

if <condition>:

<statements>

An example would be:

speed = 150

if speed > 110:

print("you have been pulled over for speeding")

Sometimes, you need to write an if statement with more complexity. The full structure of an if statement is as follows:

if <condition>:

<statements>

elif <condition>:

<statements>

elif <condition>:

<statements>

else:

<statements>

Note that the elif and the else portions are optional for an if statement.

An example of a more complex if statement is as follows:

score = int(input())

if score >= 86:

print(“A”)

elif score >= 75:

print(“B”)

else:

print(“St George’s FAIL”)

1. if statements can be combined using and, or and not along with nesting (if statements inside of if statements). The choice when to use them is dependent on the particular problem and your own preference.
2. **Important!** Remember that the = operator means **assignment**, not equals. The equality operator in Python is written as = =. Confusing the two is the most common source of error when writing if statements.
3. Python uses what is called “short circuit evaluation” when evaluating multiple conditions. If we consider the following example

if divisor != 0 and number / divisor == 25:

# do something

When Python executes the above if statement, it first checks to see if the divisor is 0. If this is the case, then the second part of the if statement will NOT be executed.

In general, if the first condition is false the rest of the conditional will NOT be evaluated. What does short circuit evaluation mean for or statements??

1. **Boolean data types**

Boolean data types can be useful. A bool variable only has two values, true or false. Here’s an example program fragment:

raining = true;

if raining:

print(“Bring an umbrella!”);

1. **Random Numbers**

Random numbers are generated as follows:

import random (this must be written in at the top of your program)

roll = int(random.random() \* 6) + 1

which generates a random number from 1 to 6.

1. **Mod Command**

The mod command (%) gives you the remainder when you divide as illustrated below:

|  |  |
| --- | --- |
| 7 % 5 = 2 | 7 / 5 = 1 |
| 3 % 10 = 3 | 3 / 10 = 0 |
| 6 % 3 = 0 | 6 / 3 = 2 |
| 14 % 5 = 2 | 14 / 5 = 2 |

The mod command has uses, particularly in divisibility tests. For example, if we wish to determine a leap year (simplistically), we would use the following:

if year % 4 == 0:

print(“leap year”);

Note that integer division in the latest version of Python is the symbol //

1. **Programming Projects**
2. Write a short program that inputs three numbers and determines which number is in the middle. For example, if you input 4, 1, 9 then the program should print out 4.
3. Write a program that determines a speeder’s fine given the following table:

|  |  |
| --- | --- |
| km/h over the limit | Fine |
|  |  |
| 1 to 20 | $100 |
| 21 to 30 | $270 |
| 31 or above | $500 |
|  |  |

You should input the observed speed of the vehicle but you may assume that the speed limit is fixed (i.e., 60 km/h or whatever else you choose)

1. Write a short program that determines if a is a multiple of b.
2. A leap year is defined as any year divisible by 4 but not by 100 UNLESS that year is divisible by 400– then it is a leap year as well. So, for example, 2000 was a leap year but 1900 was not. Write a program which prints out whether a given year is a leap year or not.