Computer Programming for Non-Majors

Lecture #1 - Getting Started: An Introduction to Programming in Python

What Is Programming?

- Computers cannot do all the wonderful things that we expect without instructions telling them what to do.
- ***Program*** – a set detailed of instructions telling a computer what to do
- ***Programming*** – designing and writing computer programs
- ***Programming language*** – a language used to express computer programs.

A First Program

```
print("This is my first Python program.")
```

Can be run as a single statement in the Python Shell or as a separate module. We'll call this **`myfirst.py`**

A First Program – What Does It Do?

```
print("This is my first Python program.")
```

Prints the message

```
This is my first Python program.
```

Ends at the end of the line

A second program

Problem – write a program which can find the average of three numbers.

Let's list the steps that our program must perform to do this:

- Add up these values
- Divide the sum by the number of values
- Print the result

Each of these steps will be a different statement.

Writing Our Second Program

- Add up these values → **`sum = 2 + 4 + 6`**
- Divide the sum by the number of values
- Print the result

`sum = 2 + 4 + 6` ← an assignment statement

Assignment Statements

- Assignment statements take the form:

  variable **=** *expression*

  Memory location where the value is stored

  Combination of constants and variables

Expressions

- Expressions combine values using one of several ***operations***.
- The operations being used is indicated by the ***operator***:

+	Addition
-	Subtraction
*	Multiplication
/	Division

Expressions – Some Examples

```
2  + 5
4  *  value
x  /  y
```

Writing Our Second Program

- `sum = 2 + 4 + 6`
- Divide the sum by the number of values
- Print the result

`average = sum / 3`

Names that describe what the values represent

Writing Our Second Program

- `sum = 2 + 4 + 6`
- `average = sum / 3`
- Print the result

```
print('The average is ',average)
```

The output method

variable name

Writing Our Second Program

```
sum = 2 + 4 + 6
average = sum / 3
print('The average is ', average)
```

Variables and Identifiers

- Variables have names – we call these names ***identifiers***.
- Identifiers identify various elements of a program (so far the only such element are the variables.
- Some identifiers are standard (such as **`sqrt`**)

Identifier Rules

- An identifier must begin with a letter or an underscore _
- Java is case sensitive upper case (capital) or lower case letters are considered different characters. **`Average`**, **`average`** and **`AVERAGE`** are three different identifiers.
- Numbers can also appear after the first character.
- Identifiers can be as long as you want but names that are too long usually are too cumbersome.
- Identifiers cannot be reserved words (special words like **`if`**, **`while`**, **`class`**, etc.)

Some Illegal Identifiers

Illegal Identifier	Reason	Suggested Identifier
`my age`	Blanks are not allowed	**`myAge`**
`2times`	Cannot begin with a number	**`times2`** or **`twoTimes`**
`four*five`	* is not allowed	**`fourTimesFive`**
`time&ahalf`	& is not allowed	**`timeAndAHalf`**

Another Version of Average

- Let's rewrite the average program so it can find the average any 3 numbers we try:
- We now need to:
  1. Find our three values
  2. Add the values
  3. Divide the sum by 3
  4. Print the result

Writing Average3a

This first step becomes:

1.1 Find the first value

1.2 Find the second value

1.3 Find the third value

2. Add the values
3. Divide the sum by 3
4. Print the result

Writing Avg3 (continued)

Since we want the computer to print out some kind of prompt, the first step becomes:

1.1.1 Prompt the user for the first value

1.1.2 Read in the first value

1.2.1 Prompt the user for the second value

1.2.2 Read in the second value

1.3.1 Prompt the user for the third value

1.3.2 Read in the third value

2. Add the values
3. Divide the sum by 3
4. Print the result

Writing Avg3 (continued)

We can prompt the user with:

1.1.1 `println("Enter the first value ?")`

1.1.2 Read in the first value

1.2.1 `print("Enter the second value ?")`

1.2.2 Read in the second value

1.3.1 `print("Enter the third value ?");`

1.3.2 Read in the third value

2.Add the values

3. Divide the sum by 3

4. Print the result

The `input` Function

- Most programs will need some form of input.
- At the beginning, all of our input will come from the keyboard.
- To read in a value, we need to use the `input` function, which will gives us the input as a string of characters.

Character Strings

- We are usually interested in manipulating sets of characters, what we call character strings.
- We can store more than one character by writing:

```
s = "This is a string."
```

- For now, we use character data for input and output only.

A Program That Uses A String Variable

```
name = input("What is your name?")
print("Glad to meet you, ",  name)
```

Writing the input statements in **Average3a**

We can read in a value by writing:

```
value1 = int(input("What is the first value?"))
value2 = int(input("What is the second value?"))
value3 = int(input("What is the third value?");
```

2. Add the values
3. Divide the sum by 3
4. Print the result

Writing the assignments statements in Average3a

```
value1 = int(input("What is the first value?"))
value2 = int(input("What is the second value?"))
value3 = int(input("What is the third value?"))
```

```
sum = value1 + value2 + value3
```

3. Divide the sum by 3
4. Print the result

Adding up the three values

Writing the assignments statements in Average3a

```
value1 = int(input("What is the first value?"))
value2 = int(input("What is the second value?"))
value3 = int(input("What is the third value?"))
sum = value1 + value2 + value3
average = sum / 3
```

4. Print the result

Calculating the average

Writing the output statement in Average3a

```
value1 = int(input("What is the first value?"))
value2 = int(input("What is the second value?"))
value3 = int(input("What is the third value?"))
sum = value1 + value2 + value3
average = sum / 3
print("The average is ", average)
```

```
value1 = int(input("What is the first value?"))
value2 = int(input("What is the second value?"))
value3 = int(input("What is the third value?"))
sum = value1 + value2 + value3
average = sum / 3
print("The average is ", average)
```

Another example – calculating a payroll

- We are going to write a program which calculates the gross pay for someone earning an hourly wage.
- We need two pieces of information:
  - the hourly rate of pay
  - the number of hours worked.
- We are expected to produce one output: the gross pay, which we can find by calculating:
  - Gross pay = Rate of pay * Hours Worked

Our Design for payroll

1. Get the inputs
2. Calculate the gross pay
3. Print the gross pay

We can substitute:

1.1 Get the rate

1.2 Get the hours

Coding the payroll program

- Before we code the payroll program, we recognize that the values (**`rate`**, **`hours`** and **`gross`**) may ***not*** necessarily be integers.
- We will convert declare these to float when we get the values as input, which means that they can have (but ***do not have to*** have) fractional parts.

Developing The Payroll Program (continued)

1.1 Get the rate

1.2 Get the hours

2. Calculate the gross pay
3. Print the gross pay

```
rate = float(
         input("What is your hourly pay rate?"))
```

Developing The Payroll Program (continued)

```
rate = float(
       input("What is your hourly pay rate?"))
```

1.2 Get the hours

2. Calculate the gross pay
3. Print the gross pay

```
hours = float("How many hours did you work?"))
```

Developing The Payroll Program (continued)

```
rate = float(
       input("What is your hourly pay rate?"))
hours = float("How many hours did you work?"))
```

2. Calculate the gross pay
3. Print the gross pay

```
gross = rate * hours
```

Developing The Payroll Program (continued)

```
rate = float(
       input("What is your hourly pay rate?"))
hours = float("How many hours did you work?"))
gross = rate * hours
```

3. Print the gross pay

```
print("Your gross pay is $", gross);
```

payroll.py

```
rate = float(
         input("What is your hourly pay rate?"))

hours = float("How many hours did you work?"))

gross = rate * hours
print("Your gross pay is $", gross);
```

Comments

- Our program is a bit longer than our previous programs and if we did not know how to calculate gross pay, we might not be able to determine this from the program alone.
- It is helpful as programs get much longer to be able to insert text that explains how the program works. These are called ***comments***. Comments are meant for the human reader, not for the computer.
- In Python, anything on a line after a double slash (**#**) is considered a comment.

payroll.py

```
# This program calculates the gross pay for an
# hourly worker.
# Inputs - hourly pay rate and number of hours
# worked
# Output - Gross pay

# Get the hourly pay rate
rate = float(
         input("What is your hourly pay rate?"))

# Get the number of hours worked
hours = float("How many hours did you work?"))

#  Calculate and display the gross pay
gross = rate * hours
print("Your gross pay is $", gross);
```

Using Stepwise Refinement to Design a Program

- You should noticed that when we write a program, we start by describing the steps that our program must perform and we subsequently refine this into a long series of more detailed steps until we are writing individual steps. This is called ***stepwise refinement***.
- Stepwise refinement is one of the most basic methods for developing a program.

Example – A program to convert pounds to kilograms

- Our program will convert a weight expressed in pounds into kilograms.
  - Our input is the weight in pounds.
  - Our output is the weight in kilograms
  - We also know that

    Kilograms = Pounds / 2.2

Pounds to Kilograms Program (continued)

- Our program must:
  1. Get the weight in pounds
  2. Calculate the weight in kilograms
  3. Print the weight in kilograms

Pounds to Kilograms Program (continued)

1. Get the weight in pounds
2. Calculate the weight in kilograms
3. Print the weight in kilograms

```
lbs = float(input
      ("What is the weight in pounds?"))
```

Pounds to Kilograms Program (continued)

```
lbs = float(input
      ("What is the weight in pounds?"))
```

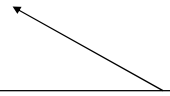
2. Calculate the weight in kilograms
3. Print the weight in kilograms

```
kg = lbs / 2.2
```

Pounds to Kilograms Program (continued)

```
lbs = float(input
      ("What is the weight in pounds?"))
kg = lbs / 2.2
```

3. Print the weight in kilograms



```
print("The weight is ", kg, " kilograms")
```

ConvertPounds.py

```
# Convert pounds to kilograms
# Input - weight in pounds
# Output - weight in kilograms

# Get the weight in pounds
lbs = float(input("What is the weight in pounds?"))

# Calculate and display the weight in
# kilograms
kg = lbs / 2.2
print("The weight is ", kg, " kilograms")
```

Another Example – The Area of A Rectangle

- Our program will calculate the area of a rectangle.
  - Our input is the length and width.
  - Our output is the area.
  - We also know that

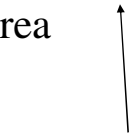
    Area = Length * Width

Our Program's Steps

1. Find the length and width
2. Calculate the area
3. Print the area

Our Program's Steps (continued)

1. Find the length and width
2. Calculate the area
3. Print the area



1.1 Find the length
1.2 Find the width

Our Program's Steps (continued)

1.1 Find the length
1.2 Find the width
2. Calculate the area
3. Print the area

```
length = float(input("Enter the length?"))
width = float(input("Enter the width?"))
```

Our Program's Steps (continued)

```
length = float(input("Enter the length?"))
width = float(input("Enter the width?"))
```

2. Calculate the area

3. Print the area

```
area = length * width
```

Our Program's Steps (continued)

```
length = float(input("Enter the length?"))
width = float(input("Enter the width?"))
area = length * width
```

3. Print the area

```
print("The area is ", area)
```

RectArea.py

```
# Calculates the area of a rectangle
# Inputs - The length and width of the rectangle
# Output - The area of the rectangle

# Print an explanatory message for the user
print("Given the width and length of a
rectangle")
print("this program calculates its area." )

# Get the inputs
length = float(input("Enter the length?"))
width = float(input("Enter the width?"))

# Calculate and display the area
area = length * width
print("The area is ", area)
```