Introduction
Chapter 0

CS112
George Mason University
# Introduction

- A program is a set of instructions, for a computer that is able to perform the actions listed below.

- A computer understands how to perform the actions and calculations specified by a program on specific types.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Calculations</th>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>store a value</td>
<td>+ -</td>
<td>integers</td>
</tr>
<tr>
<td>retrieve a stored value</td>
<td>x /</td>
<td>rational numbers</td>
</tr>
<tr>
<td>get a value from the user</td>
<td>(integer division)</td>
<td>strings</td>
</tr>
<tr>
<td>return a value to the user</td>
<td>% (modulus)</td>
<td></td>
</tr>
<tr>
<td>perform a calculation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Programming Languages: Python

• The textbook is written in pseudocode, which is not a language the computer understands

• To run your code on a computer, you need to write it in a programming language that your computer understands
  – In this class, we will use Python
  – However, Python is *not important*; it is the concepts we learn that are important!
A value is stored in memory by using the **assignment operator**, `=`
- The left side is the label for a memory location, called a **variable**
- The right side is always a value (or something that simplifies to a value), that is stored in memory
- `number1` is a variable that stores a value
- for example, `number1 = 3` would assign 3 to a memory location called `number1`
Exercises

• Exercises are found in the back of every chapter
• You can complete them online, or offline
  – if you want to do them offline, you’ll need to have installed Python on your computer
• See the webpage for each book chapter for instructions on how to complete, run, and check your answers
• You’ll need to know a tiny bit of Python for these exercises, which we will introduce next
Multiplication and Division

In python, multiplication is represented with * for example, 3 * 5 will yield 15 (if you use \texttt{x} or \texttt{X}, it will think that is a memory location)

In python, integer division is represented with \textbackslash / or \textbackslash// for example, 7 / 3 will yield 2 (surprising!) for example, 7 // 3 will yield 2

In python, normal division is represented with \texttt{x/\text{(y*1.0)}} for example, 7 / (3*1.0) will yield 2.33333333
Mixing Types

In python, numbers can be added as you would expect:
  for example, \(3 + 5\) will yield \(8\)
  for example, \(3 + 5.3\) will yield \(8.3\)

In python, strings can also be added:
  for example, "much" + "fun" will yield "muchfun"

In python, you can’t mix types for addition (and other operations):
  for example, \(3 + \text{"fun"}\) will cause an ERROR

You can add together numbers and strings, but you must convert the number to a string before adding it to the string, with \texttt{str}():
  for example, \texttt{str(3) + \"fun\"} will yield \"3fun\"
Errors

• If we make a mistake in our code, it might crash or print out the wrong value
• Let’s modify one of the templates from this chapter in a bad way:

```
Drs-MacBook-Air:Chapter_0 drcica$ python read_in_number.py
Traceback (most recent call last):
  File "read_in_number.py", line 12, in <module>
    print template(5) #this calls your code with 5 as the number provided by the user and prints it
  File "read_in_number.py", line 7, in template
    return plusTe #this returns the sum to the user
NameError: global name 'plusTe' is not defined
Drs-MacBook-Air:Chapter_0 drcica$
```
Errors continued

• You should use Google to look up what the errors mean (the message on the last line)
• The trace below shows the line numbers that were called when this error was encountered

```
Drs-MacBook-Air:Chapter_0 drcica$ python read_in_number.py
Traceback (most recent call last):
  File "read_in_number.py", line 12, in <module>
    print template(5)  #this calls your code with 5 as the number provided by the user and prints it
  File "read_in_number.py", line 7, in template
    return plusTe  #this returns the sum to the user
NameError: global name 'plusTe' is not defined
Drs-MacBook-Air:Chapter_0 drcica$  
```
A value is stored in memory by using the assignment operator, =

• The left side is the label for a memory location, called a variable
• The right side is always a value (or something that simplifies to a value), that is stored in memory
• number1 is a variable that stores a value
• for example, number1 = 3 would assign 3 to a memory location called number1
Variables

• Variables are names *you choose* for memory locations
• You pick their names from letters, numbers, or the underscore (no spaces!)
  – cannot start with a number
  – Python is case-sensitive, so `number1` and `Number1` are different variables
• Descriptive variable names are preferred to single letters
  – so `difference` is better than `diff`, `d`, or `a`
Expressions

- Python simplifies expressions, like `number1 + 10`, to be used in an assignment or return statement.
- Like mathematical expressions, precedence matters; use parentheses as needed.
  - `3 + 2 * 5` is different than `(3 + 2) * 5`
Expressions and types

- Python cannot mix string and numeric types in expressions
  - convert a number to a string using `str()` for example, `str(3) + "fun"`
  - convert a string to an integer with `int()` or rational number with `float()` for example, `int("3") + 5`
Comments

**Comments** start with `#` and are not run
- They can occur on the same line as code; anything to the right of `#` does not get run
- You can put English notes inside your code like this

```python
def template(n1):
    number1 = n1 # this gets the number from the user
    plusTen = number1 + 10
    return plusTen

# END OF YOUR CODE
print template(5) # this calls your code with 5 as
What else is in this example?

- Python cares about indentation, but we will learn why later.
- Your text editor will highlight variables, numbers, reserved words (keywords), strings and comments in different colors.
- **Keywords** have special meaning in python, and cannot be used as variable names; your text editor will highlight them in blue.
- We will learn about the first and last lines later.