Programming Using Numbers (part 1)

CMSC 104 Section 2
September 16, 2025

Administrative Notes

Don’t forget - Classwork 2 due tomorrow night before midnight

Project 1 will be assigned today (two weeks to do it)

Classwork 3 will be assigned Thursday

The first quiz is next Thursday, September 25

- It will cover everything up through this Thursday’s lecture
- Next Tuesday’s lecture WILL NOT be covered on this quiz!
- There will be a sample exam made available this Thursday to let you become familiar with the quizzes
- We will go over it in class next Tuesday
- For those with accommodations, we will get the quiz to SDS in time for your scheduled appointments

The python “Main program” and functions

We’re jumping ahead a little bit, but I need to make sure you’re aware of the terms “main program” and “function”

When a Python program starts executing: the interpreter looks for the start of the main program

- The first instruction in the main program is the first instruction that gets executed by the computer
- Functions only get executed when you call them - later in the semester

If there’s no explicit marking of “main program” and “function” the Python interpreter assumes the whole program is the main program

A digression - main programs and functions

The textbook is obsolete `def main() went out a few years ago`

- Here is how Python handles the main program currently:

```
if __name__ == “__main__”: #there are four pairs of underscores
    # the rest of the main program should be indented
```

So far we haven’t written any functions, but there’s no harm getting in the habit of identifying where your main program starts

So we will generally start the executable parts of our programs with the line above

Integers vs real numbers

Python recognizes two different types of numbers;

- Integers - whole numbers; numbers without a decimal place or an exponent
  - Positive, negative or zero
  - Python as a language does recognize a “maximum integer” - it depends on what your computer can store
- Real numbers - have decimal places and/or exponents
  - 6.02e23 or 6.02E23 - in this once instance Python is not case sensitive

Valid operations for Integers

Operator	What it does	example
+	addition	5 + 6 = 11
-	subtraction	13 - 9 = 4
*	multiplication	5 * 8 = 40
**	exponentiation	2**4 = 16
//	Integer division - yields only the integer quotient, not the remainder	4//2 = 2 5//2 = 2
%	Modulus - yields only the remainder of integer division, not the quotient	4%2 = 0 5%2 = 1

Valid operations for floating point (real) numbers

Operator	What it does	example
+	addition	5.2 + 6.3 = 11.5
-	subtraction	13.8 - 9 = 4.8
*	multiplication	5.3 * 8 = 42.4
**	exponentiation	2.2**4 = 23.4256
/	Floating point division	5/2 = 2.5

Some examples

Order of operations

What is the value of the expression:

```
5+9*3**2-7//4
```

The answer is 85. But how did we get that?

The rules Python enforces are similar to what you learned in math:

- Anything in Parentheses comes first. Start with the innermost parentheses, and work out. Go left to right if there are multiple, separate sets of parentheses
-

Order of operations:

After you've handled all parentheses:

- Next comes exponentiation.
    - If you have multiple exponents go ***right to left.*** That is,

4**3**2 is 4**9 = 262144. It's NOT 64**2 = 4096

- Then comes multiplication and division.
    - Go ***left to right*** in the expression
    - Both integer division (//) and modulus (%) count as "division"
- Last comes addition and subtraction
    - Again, go ***left to right*** in the expression

Casting variables - changing the type of a variable

Automatic conversion by the system - int to floating point

- Let’s look at what happens with 5/2 and 5//2
- The difference is that with floating point division, Python automatically casts - changes the type of - the result to floating point

Casting using int(), float(), and str()

- You can also force a cast - a change in type - yourself

Doing math with input from the keyboard

Remember from last week that all input from the keyboard is read as a string

- You can’t do math on strings in Python!!
- So you have to cast the variable you read in to an integer or a floating point, as is needed

```python
# ask the user to input a positive integer
#find the sum of all the numbers between 1 and the
number that the user entered
sum = 0
print("Welcome to my program. Enter a positive
integer ")
print(" and we will print out the sum off all the
positive integers")
print("up to and including your number")
num = input("Please enter a positive integer: ")
for index in range(1,num+1,1):
   sum = sum + index
print("The answer is ", sum)
print("Thanks for playing. Isn't this cool?")
```

```python
# ask the user to input a positive integer
#find the sum of all the numbers between 1 and the
number that the user entered
sum = 0
print("Welcome to my program. Enter a positive integer
")
print(" and we will print out the sum off all the
positive integers")
print("up to and including your number")
num = int(input("Please enter a positive integer: "))
for index in range(1,num+1,1):
   sum = sum + index
print("The answer is ", sum)
print("Thanks for playing. Isn't this cool?")
```