15-112 Fundamentals of Programming

Lecture 4 – Language basics

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Examples

- print (3 * 2)
- print (3 + 2)
- print ("abc" + "def")
- print (3 + "def")
- print 2+3*4
- print 9**1/2
- print 9**1//2
- print ("20/3 =", (20//3))
- print (" 6/3 =", (6/3))

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

More Examples

- a = 5
  print (a)
- print (5 < 8)
- print (8 < 5)
- print (8 == 8)

More Examples

- print (8 != 8)
- a = 5
  b = 6
  print (a < b)
- print (5 / 0)
- print (0 / 5)

Variables in Expressions

- Assign value to a variable
  - age = 21
- Change a variables value

```
age = 21
print ("You are "  , age * 12 , " months old")
age = age + 1
print ("You will be ", age * 12, "months after 1 year")
```


Variables in Expressions

```
radius = 3.1
pi = 22/7
area = pi * radius**2
print (area)
```

Operations

- Bitwise operators
  - & (Bitwise AND)
  - | (Bitwise OR)
  - ^ (Bitwise XOR)
  - <<
  - >>

Bitwise Operators: Examples

- **6 & 5**
- **6 | 5**
- **6 ^ 5**
- **6 << 1**
- **6 << 2**
- **6 >> 1**

More Examples

- print (1 << 2)
- a = 5
  print (a & 4)
- print (5 ^ 7)

Let's work out a problem

- Write a program that reads current temperature from the user in Fahrenheit and prints the equivalent Celsius value.

Another Example

- Write a program that reads an integer from the user and prints the sum of its digits.

Operator Precedence

- Operator precedence (highest to lowest):
  - **
  - Positive, negative, NOT (+x, -x, ~x)
  - *, /, %
  - +,-
  - >>, <<
  - & (Bitwise AND)
  - ^ (Bitwise XOR)
  - | (Bitwise OR)
- Operators with same precedence are processed left to right

Operator Precedence Examples

- print (3 + 4 * 2 + 5)
- print (3 * 2 + 2 / 5)
- print (-2 ** 4 + 8 >> 2)

Approximating Floats

What is the output of the following code?

```
d1 = 0.1 + 0.1 + 0.1
d2 = 0.3
print (d1 == d2)
```

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Short Circuit Evaluation

- Let's try the following code:

```
x = 0
y = 0
print ((y == 0) or ((x/y) == 0))
print ((y != 0) and (x/y == 0))
print (((x/y) == 0) or (y == 0))
```


Short Circuit Evaluation

- How about:

```
x = 0
y = 0
print ((y > 0) and ((x/y) == 0))
print ((y == 0) and ((x/y) == 0))
```

Strings

- Any sequence of characters enclosed within " " or ' ' is a string
  - "This is a string"
  - 'this is also a string'
  - "this is not a string – can you guess why?'
  - '7his 1s a $tring'
  - "%^%$#@!*(*&^& - what did you say?"

Indexing and Slicing

- Used to manipulate information in a string

name = "Chris Myers"

0	1	2	3	4	5	6	7	8	9	10
C	h	r	i	s		M	y	e	r	s

print name[2:4]

print name[:4]

print name [3:]

print name[:]

Math functions

- **print math.sqrt(5)** **Does not work**
- **import math**
  print math.sqrt(5)
- **math.log(x[, base])**
- **math.cos(x)**
- **math.sin(x)**
- **math.tan(x)**
- **math.pi**
- **math.e**

ord and chr functions

- ord
  - A function that will return the ASCII value of a character
- chr
  - A function to convert ASCII value to character
- Examples!

Functions

- Function is a way of packaging a group of instructions that perform a specific task
- Functions abstract out the “what” from the “how”
  - When we use a function, we worry about “What” does the function do and NOT “how” it does it.
  - When we write a function we worry about the “how”.

Functions that do something

- Some functions just perform a task

```
def doSomething():
    print("CMU Rocks!")
```

- How would you use this function

```
doSomething()
doSomething()
```

Functions that act on input

- Some functions perform a tasks on values that you give them
  - printSquare – A function that takes a number and prints its square
  - How will you use this function?

```
printSquare(2)
printSquare(3)
```

  - How will you define this function?

```
def printSquare(x):
    print x, "**2 =", (x*x)
```

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Function definition

def SomeName (Input parameters if any):
    Function Body
    Function Body
    Function Body

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Using Functions

- A function has to be defined before it can be used!
- A complete example – funtest.py

```
def printSquare(x):
    print x, "**2 =", (x*x)

printSquare(2)
printSquare(3)
```


Functions - multiple parameters

- Functions can take several parameters

```
def printSum(x,y):
    print x, "+", y, "=", x+y

printSum(2,3)
printSum(3,4)
```

Functions with return values

- Functions can return values

```
def square(x):
    return x*x

print square(3)
print square(4)
a = square(3) + square(4)
print a
```


print vs return

- What does print do?
- What does return do?

A more complex example

- Write a program that reads the number of eggs bought by a customer and based on this input, determines how many cartons of eggs the customer would need. We can fit 12 eggs in one carton.

More Exercises

- **isEvenPositiveInt(x)**
- **isLegalTriangle(s1, s2, s3)**
- **rectanglesOverlap(left1, top1, width1, height1, left2, top2, width2, height2)**