Programming Club Meeting 5 Slides

ACM Code Jam 2023

DSUACM Student Chapter
Association for Computing Machinery

==========

>>

Computer Programming Contest for all students in grades 9-12 from districts represented by Colonial and Carbon Lehigh Intermediate Units. Teams of 4 will be given a certain number of problems to solve and the team that correctly completes the most problems will win the competition.

>>

==========

>>

>>When: December 20, 2022 (8:30 am - 11:30 am),
Snow Date: March 10, 2023

>>Where: DeSales University,
2755 Station Ave, Center Valley, PA 18034

>>How: For more information and to register your team, visit www.ciu20.org/codejam/

>>Registration Deadline: December 1, 2022
Refreshments will be provided.
Lunch will be on your own.

>>

==========

>>

>>Points of Contact:

>>

>>Heather Heimer (IU #20) hheimer@ciu20.org

>>Diahann Ouly (IU #21) oulyd@cliu.org

>>

==========

COLONIAL Intermediate Unit 20
Colonial IU #20

Carbon Lehigh IU #21

DeSales University

Meeting 1

Data Types

Code

```
1  # Ints, floats, bools, chars, strings
2  i = 7_200_00 # integer, int()
3  f = 2_110.100 # float, double, float()
4  b = True # boolean, bool()
5  c = 'Q' # char, chr()
6  s = "hello world!" # string, str()
7  print(type(f))
8
9  print()
10
11 # Types don't mix
12 # print("Hello " + 5) # causes error, doesn't know what you want
13 print(f"Hello {5}") # only in Python 3
```

Output

```
<class 'float'>

Hello 5
```

```
Line 12: TypeError: cannot concatenate 'str' and 'int' objects
```

Math and Comparison

- Math:
  - Normal operators (+, -, *, /)
  - Weird operators (on right)
- Comparison
  - Normal comparison (>, <)
  - >=, <=
  - Equals and doesn't equal
  - And, or, and not

Code

```
# Math Operators
print(5 // 3) # floor division (divides then rounds down)
print(5 % 3) # modulus, remainder in division
print(5 ** 3) # exponent
from math import sqrt
print(sqrt(5))

print()

# Comparison
print(3 <= 5)
print(3 == 5)
print(3 != 5)
print(True and False)
print(False or True)
print(not(True))
```

Output

```
1
2
125
2.23606797749979

True
False
True
False
True
False
```

Meeting 2

If, Else, Elif

Code

```
# If
b = True
if (b):
    print("True")

print()

# If Else
if (not(b)):
    print("True")
else:
    print("False")

# Elif
i = 3
if (i == 1):
    print("True")
elif (b == 3):
    print(i)
else:
    print("False")
```

Output

```
True

False
False
```

Meeting 3

Strings

- Indexing
  - Pick a specific spot [x]
  - Pick a range [x:y]
  - Reverse [::-1]
- Methods
  - W3Schools Reference Page
- Escape Characters
  - \n, \t, \\, \", \'

Code

```
1  string = "Hello world!"
2
3  # Indexing
4  print(string[0:3])
5  print(string[::-1])
6
7  print()
8
9  # Methods
10 print(string.upper())
11 print(string)
12 string = string.upper()
13 print(string)
14
15 print()
16
17 # Escape Chars
18 string = "Hello\tWorld!"
19 print(string)
```

Output

```
Hel
!dlrow olleH

HELLO WORLD!
Hello world!
HELLO WORLD!

Hello	World!
```

Loops

Code

```
1  string = "Hello"
2
3  # While Loop
4  while (len(string) < 9):
5      string += "_"
6  print(string)
7  print("123456789")
8
9  print()
10
11 # For Loop
12 for i in range(2,7):
13     print(i)
```

Output

```
Hello____
123456789

2
3
4
5
6
```

Meeting 4

Functions

Code

```
1  string = "Hello"
2  def myFunc(var: str = "test") -> str:
3      print(string) # can access global
4      print(var)
5      funcVar = 32
6      return var*2
7
8  print(myFunc("World"))
9  print()
10 # print(funcVar) # can't access function var
11 print(myFunc(string))
12 print()
13 print(myFunc())
```

Output

```
Hello
World
WorldWorld

Hello
Hello
HelloHello

Hello
test
testtest
```

Example of Splitting Into Subparts / Pseudocode:

```
Read the first character
while not the end of input data
{
    a. initialize the stack
    b. process the expression
    c. output result
    d. get the next expression
}
while (ch is not = '=') //process each expression
                        //= marks the end of an expression
{
    switch (ch)
    {
    case '#':
        read a number
        output the number;
        push the number onto the stack;
        break;
     default:
        assume that ch is an operation
        evaluate the operation;
    } //end switch

        if no error was found, then
        {
            read next ch;
            output ch;
        }
        else
            Discard the expression
    } //end while
```

Practice

Example Problem: asteriskGenerator

- Src: (Number 7) https://ashaicy99.medium.com/python-nested-for-loops-practice-exercises-dee4e76a00bb
- Goal: Write a Python program which will output the shown asterisk pattern without keeping a string for multiple lines (use nested loops).
- The program should work for any number of asterisks in the middle (longest) row.
- Relevant Information:
  - Should probably make ‘asteriskGenerator’ function
  - Use nested loops instead of just adding or subtracting asterisk from string
  - Should have user input

Pattern:

```
*
* *
* * *
* * * *
* * *
* *
*
```

Practice Problem 1: maxNum Function

- Goal: Write a Python function that finds the max of three inputted numbers.
- Also write a program that shows this functionality.
- Relevant Information:
  - Variable names like “max” or “min” should be avoided because they already exist in Python

Practice Problem 2: caseCalc Function

- Goal: Write a Python function that will determine the number of upper and lower case letters in a string.
- Also write a program that will test this functionality.
- Relevant Information:
  - You can print the number of upper and lower case letters
  - Or you can use “return (lower, upper)” with the call to function being: “lower, upper = caseCalc(string)”

Practice Problem 3: isPrime Function

- Goal: Write a Python function that will determine if the inputted number is prime or not.
- Also write a program that will test this functionality.
- Relevant Information:
  - A prime number is a number which is only divisible by 1 and itself.
  - Tests -
    - Prime: 2, 13, 73, 97
    - Composite: 6, 8, 56, 63
    - Other: 1