Computer Science 125
8 September 2020

1. What is a boolean expression?

Statement that outputs True or False.

Boolean values

2. How are comparison operators different from logical operators?

<, >, <=, >=

==, !=

and, or, not

3. What are the logical opposites of the following expressions?

`a > 5`

not (a > 5)

a <= 5

`a == 0 or b >= 5`

not (a == 0 or b >= 5)

a != 0 and b < 5

not (A and B) = (not A) or (not B)

not (A or B) = (not A) and (not B)

a	b	a==0 or b>=5	a!=0 and b<5
0	4	True True	F T False
1	4	F F False	T T True

4. What does the following program do?

```
n = int(input("Enter an integer:"))
if n%2 == 0:
  print("yes")
else:
  print("no")
```

$$n \% 2 = \begin{cases} 0 & \text{if } n \text{ even} \\ 1 & \text{if } n \text{ odd} \end{cases}$$

Determines whether or not n is even.

5. **Write a function isVowel(c) that accepts a lowerchase letter c, and returns True if c is a vowel and False otherwise.**

Assume c is a string, contains a letter

c = "a"
c = "g"

```
def isVowel(c):
    if c=="a" or c =="e" or c=="i" or c=="o" or c=="u":
        return True        ends the function
    ~~else~~:
    return False
```

~~c==("a", "e", "i"...)~~

```
Test cases:    isVowel("a")    #should return True
               isVowel("u")
               isVowel("f")    # should return False

isVowel(5)
```

6. **Write a function classTime(day) that accepts the name of a weekday and prints the time that CS 125 class starts on that day.**

day: "Tuesday"

```
def classTime (day):
    if day == "Tuesday" :
        print ("class starts at 11:45 am")
    elif day == "Thursday" :
        print ("class starts at 12:45 pm")
    else:
        print("class does not meet today")
```

```
Test cases:    classTime ("Wednesday")
```

PRACTICE WITH SELECTION – SOLUTIONS

CS 125

Working with a partner/group, use the following steps to solve each of the following problems.

(a) Plan your function on the white board (either on the classroom wall or on Zoom). Write out your entire program. Think about what errors might occur and how to fix them.

(b) Plan multiple test cases for your function. What input will you send to your function? What value should the function return?

(c) *Only after you have completed steps (a) and (b) should you type your code in Python.*

(d) After you have typed your function, run your test cases. Does your function work? If not, how can you fix it?

Here are the practice problems:

1. Write a function `decimalToPercent(d)` that accepts a number *d* as a parameter. If the number is between 0 and 1 (inclusive), then your function prints the number as a percent (that is, multiply *d* by 100 and print a percent sign). Otherwise, the program prints the message "You entered a number that is not between 0 and 1."

```
def decimalToPercent(d):
    if 0 <= d and d <= 1:
        print(d*100, "%")
    else:
        print("You entered a number that is not between 0 and 1.")
```

2. Write a function `triangleOrSquare()` that prompts the user to enter either "triangle" or "square". If the user enters "triangle", then your function draws a triangle. If the user enters "square", you're your function draws a square. Otherwise, your function prints an error message.

```
def drawSquareOrTriangle(anyTurtle):
    flag = input("Enter triangle or square:")
    if flag == "triangle":
        for i in range(3):
            anyTurtle.forward(50)
            anyTurtle.left(120)
    elif flag == "square":
        for i in range(4):
            anyTurtle.forward(50)
            anyTurtle.left(90)
    else:
        print("error: invalid choice")
```

3. Write a function `maxNum()` that prompts the user to enter three numbers and returns the largest of the three numbers.

```
def maxNum():
    a = float(input("Enter a number:"))
    b = float(input("Enter a number:"))
    c = float(input("Enter a number:"))

    if a >= b and a >= c:
        return a
    if b >= a and b >= c:
        return b
    return c
```

4. Write a function `weekday()` that asks the user for the name of a weekday, such as `"Monday"` or `"Thursday"`. If the user does not enter a valid day name, then print an error message. Otherwise, then tell the user whether *today* is the day of the week that they entered. To do this, use the `datetime` module to get the current day of the week as follows:

```
import datetime
dayOfWeek = datetime.date.today().strftime(“%A”)
```

```
def weekday():
    day = input("Enter the name of a weekday:")
    if (day != "Sunday" and day != "Monday" and day != "Tuesday"
        and day != "Wednesday" and day != "Thursday"
        and day != "Friday" and day != "Saturday"):
        print("Please enter a valid day.")
    else:
        #find out what day it is from the datetime module
        dayOfWeek = datetime.date.today().strftime("%A")

        if(day == dayOfWeek):
            print("Today is",day)
        else:
            print("Today is not",day)
```

5. Write a program `isPrime(n)` that takes a positive integer `n` and returns `True` if `n` is prime and `False` otherwise.

```
def isPrime(n):
    if n == 1:
        return False
    s = math.floor(math.sqrt(n)) # this is not necessary, but it
                                 #   makes the program more efficient
    for i in range(2,s+1):
        if n % i == 0:
            return False
    return True
```