

Homework 2 - Conditionals

CS 1301 - Intro to Computing - Fall 2024

Important

- Due Date: **Thursday, September 5th, 11:59 PM.**
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
 - TA Helpdesk
 - Email TA's or use class Ed Discussion
 - [How to Think Like a Computer Scientist](#)
 - [CS 1301 YouTube Channel](#)
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

HW02 will test your understanding of conditional blocks, boolean variables and expressions, and print vs. return. This homework consists of 5 functions for you to implement. You have been given a [HW02.py](#) skeleton file to fill out. You will find more detailed information below to complete your assignment. Read it thoroughly before you begin.

Hidden Test Cases: In an effort to encourage debugging and writing robust code, we will be including hidden test cases on Gradescope for some functions. You will not be able to see the input or output to these cases. Below is an example output from a failed hidden test case:

```
Test failed: False is not true
```

Written by: Maya Blair, Angelina Zhang, & Paige Holland

Helpful Information To Know

Print vs. Return

Two concepts that may be difficult for programmers to differentiate are the print function and the return statement. It is imperative to understand that printing a value and returning a value could not be more different.

- **The Print Function:** The purpose of `print()` is to display information to the user.
- **The Return Statement:** The return statement is the "output" of a function. Once Python hits a return statement, Python immediately exits the function back to where the function was called. The function call is then replaced with the return value, which can be used in the rest of the program.

Functions that do not explicitly have a return statement always return the value `None`. For example, let's say we have the following two functions below:

```
def printFunc():  
    print(2)  
def returnFunc():  
    return 2
```

This is what would happen if we typed the following into the shell:

```
>>> a = printFunc()  
2  
>>> print(a)  
None
```

Note that although `2` is printed to the screen, `a` has the value `None`, since `printFunc()` has no return value.

```
>>> b = returnFunc()  
>>> print(b)  
2
```

When we call `returnFunc()` and assign it to the variable `b`, nothing is printed to the screen because there are no print statements inside the function. However the variable `b` now holds the returned value of `2`.

Aurora's Fate

Function Name: `auroraFate()`

Parameters: `age (int)`, `isSunset (bool)`

Returns: `birthdayFate (str)`

Description: Aurora has just been cursed by Maleficent to prick her finger on a spinning wheel. Flora, one of the three good fairies, wants to look into the future to see how she can prevent the princess's fate.

Write a function called `auroraFate()` that takes in two parameters: Aurora's age (`int`) and whether it is before sunset (`bool`). If Aurora's age is 16 **and** it is sunset, then return the string "Oh no! We must find Prince Phillip.". If Aurora is **older** than 16, then return the string "Aurora is safe from the evil fairy!". Otherwise, return the string "It is too early to tell!".

```
>>> auroraFate(16, True)
"Oh no! We must find Prince Phillip."
```

```
>>> auroraFate(16, False)
"It is too early to tell!"
```

Scuttle's Human Objects

Function Name: identifyObjects()

Parameters: material (str), weight (float)

Returns: objectName (str)

Description: Scuttle and Ariel need your help identifying some human objects!

Given the table below, write a function called identifyObjects() that takes in two parameters: the material of the item (str) and weight of the item in pounds (float).

Material	Weight (lbs)	Object
Metal	0.5	Dinglehopper
Metal	1.0	Snarfblatt
Glass	5.0	Thingamabob
Leather	2.0	Whosits

Return the name of the corresponding object that matches the material and weight description. If no item matched the given description, then return the string "Well then it must be a whatsits!" .

```
>>> identifyObjects("Metal", .5)
"Dinglehopper"
```

```
>>> identifyObjects("Leather", 2.0)
"Whosits"
```

Finding Cinderella

Function Name: `cinderellaSearch()`

Parameters: `candidateName` (`str`), `shoeSize` (`int`)

Returns: `suitableStr` (`str`)

Description: Cinderella has made her hasty escape from Prince Charming's magnificent ball. The prince wants to get her back and needs to quickly sort through a long list of shoe sizes to eliminate it down to a few potential candidates.

Write a function that takes in two parameters: the candidate name (`str`) and their shoe size (`int`). If the candidate's name is "Jaq" or "Gus" , only return the string "This isn't Cinderella!" . If the shoe size is between sizes 3 and 6 (inclusive), then **print** the name of the candidate and return the string "Cinderella may have a shoe size of {shoeSize}." . Otherwise, return the string "{candidateName} is not a suitable fit!" .

```
>>> cinderellaSearch("Drizella Tremaine", 8.0)
"Drizella Tremaine is not a suitable fit!"
```

```
>>> cinderellaSearch("Vanellope von Schweetz", 4)
Vanellope von Schweetz
"Cinderella may have a shoe size of 4."
```

Moana's Voyage

Function Name: `voyageDecision()`

Parameters: `weatherType (str)`, `timeOfDay (str)`

Returns: `None (NoneType)`

Description: Moana is planning her next voyage across the ocean, and she needs to decide the best route based on weather conditions.

Write a function called `voyageDecision()` that takes in two parameters: the weather condition (`str`) and the time of day (`str`). The weather conditions will either be `"sunny"`, `"rainy"`, or `"stormy"`, and the times of day will either be `"morning"`, `"afternoon"`, or `"night"`.

- If the weather is `"sunny"`, then the function should print `"Time to set sail!"`. However, if the time of day is `"night"`, then the function should print `"We'll leave first thing in the morning!"` instead.
- If the weather is `"rainy"`, then the function should print `"Let's wait for better weather."`. However, if the time of day is `"morning"`, then the function should print `"Time to set sail!"` instead.
- If the weather is `"stormy"`, then the function should print `"Uh oh, we need to cancel the voyage."`.

```
>>> voyageDecision("sunny", "afternoon")
"Time to set sail"
```

```
>>> voyageDecision("rainy", "night")
"Let's wait for better weather."
```

Magic Carpet Ride

Function Name: `aWholeNewWorld()`

Parameters: `numOfGuards` (`int`), `isAsleep` (`bool`)

Returns: `response` (`str`)

Description: Aladdin stopped by to take Princess Jasmine on a magic carpet ride. However, she has to sneak out to go.

Write a function called `aWholeNewWorld()` that takes in two parameters: the number of guards (`int`), and whether or not the Sultan is asleep (`bool`) to determine if she can go.

- If there are no guards on duty **and** the Sultan is asleep, then Jasmine can go and you should print the string `"Let's go!"` .
- If the Sultan is awake **and** there are no guards on duty, Jasmine can go and you should print the string `"I'll need to be back by midnight."` .
- If there are three guards or less on duty **and** the Sultan is asleep, Jasmine can go and you should print the string `"I can sneak around the guards."` .

Finally, return the string `"A dazzling place I never knew!"` if Jasmine was able to go on the magic carpet ride. Otherwise, return the string `"Where's a genie when you need them?"`

```
>>> aWholeNewWorld(0, False)
I'll need to be back by midnight.
"A dazzling place I never knew!"
```

```
>>> aWholeNewWorld(3, False)
"Where's a genie when you need them?"
```

Grading Rubric

Function	Points
auroraFate()	20
identifyObjects()	20
cinderellaSearch()	20
voyageDecision()	20
aWholeNewWorld()	20
Total	100

Provided

The `HW02.py` skeleton file has been provided to you. This is the file you will edit and implement. All instructions for what the functions should do are in this skeleton and this document.

Submission Process

For this homework, we will be using Gradescope for submissions and automatic grading. When you submit your `HW02.py` file to the appropriate assignment on Gradescope, the autograder will run automatically. The grade you see on Gradescope will be the grade you get, unless your grading TA sees signs of you trying to defeat the system in your code. You can re-submit this assignment an unlimited number of times until the deadline; just click the "Resubmit" button at the lower right-hand corner of Gradescope. You do not need to submit your `HW02.py` on Canvas.