Important

1. Due Date: **Thursday, January 26th**
2. This homework is graded out of 100 points.
3. This is an Individual Assignment. You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. Students may only collaborate with fellow students currently taking CS 1301, the TA's and the lecturer. You should not exchange code or write code for others. For individual assignments, each student must turn in a unique program. Your submission must not be substantially similar to another student's submission. Collaboration at a reasonable level will not result in substantially similar code.
4. For Help:
   - TA Helpdesk (Schedule posted on class website.)
   - Email TA's or use Piazza Forums Notes:
   - How to Think Like a Computer Scientists [http://openbookproject.net/thinkcs/python/english3e/]
   - CS 1301 Python Debugging Guide [http://www.cc.gatech.edu/classes/AY2016/cs1301_spring/CS-1301-Debugging-Guide/index.html]
5. Don’t forget to include the required collaboration statement (outlined on the syllabus). Failing to include the Collaboration Statement will result in no credit.
6. Do not wait until the last minute to do this assignment in case you run into problems.
7. **Read the entire specifications document before starting this assignment.**
8. **IF YOUR CODE CANNOT RUN BECAUSE OF AN ERROR, IT IS A 0.**

Introduction

The goal of this homework is for you to learn the difference between print and return as well as give you practice with conditionals. Part 1 consists of 4 simple functions and 1 function which uses those 4 functions. Part 2 consists of 2 functions that will test your knowledge of conditionals. Part 3 consists of 4 real world application functions. **You have been given HW2.py** to fill out with instructions in the docstrings. However, below you will find **more detailed information** to complete your assignment. Read it thoroughly before you begin. You have until **Thursday, January 26** to complete this assignment.

Print vs Return

Two concepts that may be difficult for beginner programmers to differentiate between are the print function and the return statement. While it may appear that they do the same thing, it is important to note that they **do not** do the same thing. The print function as its name states, is a function just like round or any other pre-defined and user defined functions. The **only purpose** for the **print function** is to **display information** to the user. You cannot save what you print. The return statement, on the other hand, is part of a function definition. All functions have a return statement, whether you explicitly write one or not; functions that do not explicitly have a return statement always return the value None. The **return statement** is useful because it allows you to **assign a value** to a function, thus allowing you to either save it for later, use it in a more complex expression, or print it for human consumption.

Important note to remember: print is for the user whereas return is for the program.

Part 1:

After a long and fulfilling time at Georgia Tech, you decide to spend a gap year before working at a farm. Write the following four functions to determine which crops are available in what season. **Test cases are provided at the end of the part.**

Function name: **`isSummerCrop`**
Parameters: **`crop` (`str`)**
Return value: **`True or False` (`bool`)**
Description:
Write a function that determines whether or not the parameter ***crop*** is in season. Refer to the table below to see which crops are in season during the summer. Return True if it is, False otherwise.

Function name: **`isWinterCrop`**
Parameters: **`crop` (`str`)**
Return value: **`True or False Boolean`**
Description:
Write a function that determines whether or not the parameter ***crop*** is in season. Refer to the table below to see which crops are in season during the winter. Return True if it is, False otherwise.

Function name: **`isFallCrop`**
Parameters: **`crop` (`str`)**
Return value: **`True or False` (`bool`)**
Description:
Write a function that determines whether or not the parameter ***crop*** is in season. Refer to the table below to see which crops are in season during the fall. Return True if it is, False otherwise.

Function name: **`isSpringCrop`**
Parameters: **`crop` (`str`)**
Return value: **`True or False` (`bool`)**
Description:
Write a function that determines whether or not the parameter ***crop*** is in season. Refer to the table below to see which crops are in season during the spring. Return True if it is, False otherwise.

Summer	**Winter**	**Fall**	**Spring**
"avocado"	"avocado"	"lettuce"	"avocado"
"watermelon"	"pumpkin"	"pumpkin"	"lettuce"
"plum"	"lemon"	"ginger"	"lemon"

Notes:

- The parameter **`crop`** will be a string; you are not responsible to account for different spellings or capitalizations of the crops listed above.
- Review the test cases below to confirm edge cases.

Test Cases (not an exhaustive list):

```
isSummerCrop("avocado") → True
isSummerCrop("avocadowatermelonplum") → False #You do not have to consider different spellings!
isSummerCrop("hi") →False
isWinterCrop("lettuce") →False
```

```
isSpringCrop("Avocado") → False # Note the capital A instead of a in avocado.
isSpringCrop("avocado") →True
```

One more! Use the functions you wrote above to help you with the next one!

Function name: **`marketPlace`**
Parameters: **`crop (str)`**
Return value: **`[0-4] inclusive (int)`**
Description:
Write a function that determines how many seasons the parameter **`crop`** is in season for by calling the four functions you wrote above. Return an int that represents that number. This return value should be `0, 1, 2, 3, or 4.`
Test Cases (not an exhaustive list):

```
marketPlace("avocado") → 3
marketPlace("ginger") → 1
marketPlace("apples") → 0
```

Part 2:

Function name: **`findMovie`**
Parameters: **`studio (str), director (str)`**
Return value: **`name of a movie (str)`**
Description:
Write a function called findMovie where the first parameter is studio and the second parameter is director, and returns a string representing the matching movie. The table below shows which movie to return based on the studio and the director:

studio	**director**		
	Spielberg	*Scorsese*	*Villeneuve*
Warner Bros	Ready Player One	The Departed	Prisoners
Universal	Jaws		
Paramount	Saving Private Ryan	Hugo	Arrival

Test Cases (not an exhaustive list):

```
findMovie("Warner Bros", "Scorsese") → "The Departed"
findMovie("Universal", "Scorsese") → None
```

Function name: **`watchMovie`**
Parameters: **`rating (int), time (int)`**
Return value: **`representing whether or not to watch the movie (bool)`**
Description:
Write a function called `watchMovie` where the first parameter is `rating` and the second parameter is `time`, and returns a boolean about whether or not to watch a movie. Return `True` if you are to watch a movie and `False` if you are to not watch a movie.
You will **not** watch the movie if either of the following are true:

- The rating is less than 60
- The rating is less than 80 and greater than or equal to 60, and the time is either less than 105 or greater than 140

You **will** watch the movie if either of the following are true:

- The rating is less than 80 and greater than or equal to 60, and the time is greater than or equal to 105 and less than or equal to 140 minutes
- The rating is greater than or equal to 80

Test Cases (not an exhaustive list):

```
watchMovie(45, 115) → False
watchMovie(60, 105) → True
watchMovie(80, 200) → True
watchMovie(75, 90) → False
watchMovie(95, 90) → True
```

Part 3:

Function name: **is_even**
Parameters: **number (int)**
Return value: **True or False (bool)**
Description:
Write a function called is_even that takes an integer as an argument and returns True if the argument is an **even number** and False if it is **odd.**

Function name: **is_odd**
Parameters: **number (int)**
Return value: **True or False (bool)**
Description:
Write a function called is_odd that takes an integer as an argument and returns True if the argument is an **odd number** and False if it is **even.**

Function name: **is_factor**
Parameters: **factor(int), number (int)**
Return value: **True or False (bool)**
Description:
Write a function called is_factor where the first parameter *factor* is an integer and the second parameter *number* is an integer. Your function should return True if the first argument (factor) is a factor of the second argument (number).
Test Cases (not an exhaustive list):

```
is_factor(3, 12) → True
is_factor(5, 12) → False
is_factor(25, 15) → False
```

Function name: **is_multiple**
Parameters: **multiple(int), number (int)**
Return value: **True or False (bool)**
Description:
Write a function called is_multiple where the first parameter *multiple* is an integer and the second parameter *number* is an integer. Your function should return True if the first argument (multiple) is a multiple of the second argument (number).
Test Cases (not an exhaustive list):

```
is_multiple(12, 3) → True
is_multiple(12, 5) → False
is_multiple(12, 12) → True
```

Grading Rubric

Part 1
- `isSummerCrop`: 8 points
    Correct Value is returned: 8 Points
- `isWinterCrop`: 8 points
    Correct Value is returned: 8 Points
- `isFallCrop`: 8 points
    Correct Value is returned: 8 Points
- `isSpringCrop`: 8 points
    Correct Value is returned: 8 Points
- `marketPlace`: 18 points
    Passes some test cases: 5/18 Points
    Passes all test cases: 18/18 Points

Part 2
- `findMovie`: 15 points
    Passes some test cases: 5/15 Points
    Passes all test cases: 15/15 Points
- `watchMovie`: 15 points
    Passes some test cases: 5/15 Points
    Passes all test cases: 15/15 Points

Part 3
- `is_even`: 5 points
    Passes all test cases: 5 Points
- `is_odd`: 5 points
    Passes all test cases: 5 Points
- `is_factor`: 5 points
    Passes all test cases: 5 Points
- `is_multiple`: 5 points
    Passes all test cases: 5 Points

Provided

The following file(s) have been provided to you. There are several, but you will only edit one of them:

1. `HW2.py`
   This is the file you will edit and implement. All instructions for what the methods should do are in the docstrings.

Deliverables

You must submit all of the following file(s). Please make sure the filename matches the filename(s) below. Be sure you receive the confirmation email from T-Square, and then download your uploaded files to a new folder and run them.

1. `HW2.py`