|  |  |  |
| --- | --- | --- |
| **CS 112, Foundations of CS**  **Lab 2: Lists & Decision Structures** | |  |
|  |  | |

This lab is worth 100 points. The goal for this lab is:

1. **Getting familiar with a collection data type, list.**
2. **Getting familiar with if, if-else and if- else-if decision structures in Python**

If you do not complete the lab in the time allotted, then please return to the lab in your spare time, and complete it by the **due date, which is specified on Canvas**.

**Preliminaries**

* Open Anaconda and double click on “Spyder” icon to open up the Spyder IDE.
* For this lab, create another folder, called **lab2.**

1. **Lists (Cars.py)**

For this programming task you will create 3 lists (makes, models and years) and initialize them. Then you will update one of the values in the makes and years lists by indexing them. After that you will add another car to the collection by using the append() method. At the end you will invoke a few print statements to print everything out.

1. To get started create a new python file (mouse-click on the *File* menu, then select *New file*). If you need help please ask or refer to lab 1.
2. In the comment section on top, include your name, today’s date and program name (Lab 2, Cars.py).
3. You will create three lists for this section(step 4 - 6). Remember for lists in Python we use **square brackets [ ]**. Items inside the list are separated with commas. In lecture we created a list named dogs, and populated it with dog names :

***dogs = [“toto”, “clifford”, “sparky”]***

1. Create a list and name it “**makes**”. Initialize the list with 4 car makes (Honda, Toyota, Mercedes and Ford).
2. Create another list and name it “**models**”. Initialize the list with 4 car models (Accord, Camry, C63AMG, F150).
3. Create another list (number 3) and name it “**years**”. Initialize the list with 4 years (2001, 1989, 2019, 2020).
4. At this point your program should have 3 lists, and they are initialized with car makes, models and years. Go ahead and save the file as ***Cars.py*** and run the file. If there aren’t any errors, then nothing will be outputted to your console. If there are errors, fix them before moving to the next step.
5. As mentioned during lecture, lists are **ordered** and **changeable**. This means we can access a specific item in the list and change the value. For this step you will write code to change the year number for the Toyota Camry. Change it from 1989 to 2019, without modifying the line of code where the list is created. In other words, I want you to index the location in “**years**” (using the [ ] operator), and assign 2019 to it. For example (referring back to the list of dogs example from lecture):

***dogs[0] = “winston”***

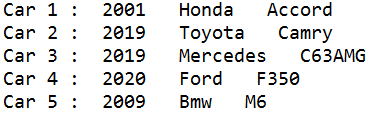
1. Similar to step number 8, where you updated the year number from 1989 to 2019, update F150 to F350 in the “**makes”** array.
2. At this point you should have 3 lists; and you updated the year number for Toyota Camry by indexing the ”**years**” list; *and* you have updated the make for Ford, from F150 to F350 by accessing the “**make**s” lists and updating values. Run the file and fix any errors if it crashed.
3. Now we will add one more car to our collection without modifying the lists where they are created. This can be accomplished by using the **append()** method. Append to the “**makes**” list BMW, to the “**models**” list M6, and to the “**years**” list 2009. If we wanted to add the dog name “bella” to our dogs list, we can do it by saying:

***dogs.append(“bella”)***

1. Print out each car make with the corresponding model and year. You should have a total of 5 print statements, as you have 5 items in each list. The first print statement can be seen below. Repeat it for the other 4.

***print(“Car 1 :”, years[0], “ ”, makes[0], “ ”, models[0])***

1. If your program is written correctly and you have no errors then the output of your program should be the following:



**II. If-Else (SeasonsAndDays.py)**

For this programing task you will write code which makes decision based on use input. You will practice using if, if-else and if-else-if (in Python known as **elif**). First you will get a number from the user and based on the value of the number, you will figure what day it is. Then you will ask the user for the season, and based on the input, you will figure out which month it is. At the end you will print the information to the screen using the print() function.

1. To get started create a new python file (mouse-click on the *File* menu, then select *New file*).If you need help please ask or refer to lab 1.
2. In the comment section on top, include your name, todays date and program name (Lab 2, SeasonsAndDays.py)
3. At times it is crucial to validate input from the user. In other words, you need to check if the user is entering a value between 1 and 10 if prompted to. If the user is entering something else, than the code can break. To avoid crashes associated with incorrect user input, you will use **sys.exit()** method. This forces a clean exit of the program. You will use sys.exit() in later steps of this program. In order to use, you need to import the system module. This can be done by saying “**import sys**”.
4. Prompt the user to enter a number between 1 and 7. Save users input into a variable named “**day\_num**”.
5. Declare two variables, **day** and **month**. Assign empty string literals to them. For example:

**day = “”.**

1. In this step you will assign a value to the variable named **day**. Write an if-elif-else which specifies what **day** it is based on the number stored in the variable **day\_num**. If **day\_num** is 1, then it is Monday. If it is 2, then is Tuesday. You should start with an **if** statement, and then have 6 more **elif**s. At the end should have an **else** clause which will cause the program to exit if the variable **day\_num** has a value which is not between 1 and 7. In the body of the else statement, print out “Incorrect day number…” and follow it with the statement **sys.exit()**. Take a look below at the the first statement for this part.

image5.png

1. Create four lists based on the table below. For example one of the lists will be named “spring”, which has three items in it: “Spring”, “spring”, “SPRING”.

|  |  |  |  |
| --- | --- | --- | --- |
| **List Name** | **Item 1** | **Item 2** | **Item 3** |
| **spring** | *“Spring”* | *“spring”* | *“SPRING”* |
| **summer** | *“Summer”* | *“summer”* | *“SUMMER”* |
| **fall** | *“Fall”* | *“fall”* | *“FALL”* |
| **winter** | *“Winter”* | *“winter”* | *“WINTER”* |

1. Save and run your file. If there are syntax errors, do fix them before continuing with other steps.
2. Prompt the user to enter “What season is it?” and store the input into the variable **season**.
3. In this step you will assign a value to the variable named **month** based on the value stored in the variable **season**. Write another if-elif-else to do this. If the season is “**spring**” than the month is “**March**”. If the season is “**winter**”, then the month is “**December**”. If the season is “**fall**”, then the month is “**September**”. If the season is “**summer**”, have another if-else inside of the body for this conditional to check if **day\_num** is less than or equal to 3. If it is then month is “**June**”, else it is “**July**”. The else clause for the outer if-elif-else for this step, should notify the user of incorrect season entry and terminate the program with **sys.exit()**.

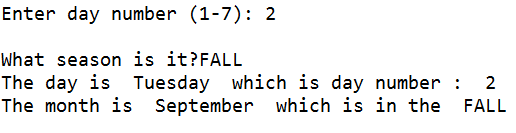
image6.pngThe user can enter the name of the season a few different ways. For example the season spring can be entered as “spring”, “SPRING”, “SPring” and so on. We know that all of these mean the same season. However, these would be completely different string literals. We can account for this by checking the different variations for each season, which we saved in our lists back at step number 7. Below is the complete code for the first part of the if-elif-else for this step.

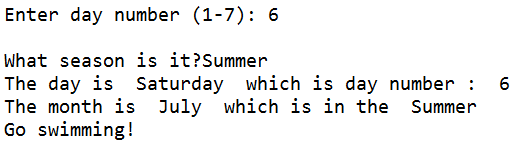
1. Invoke the print() function twice to output the information to the user. Your first print statement should be :

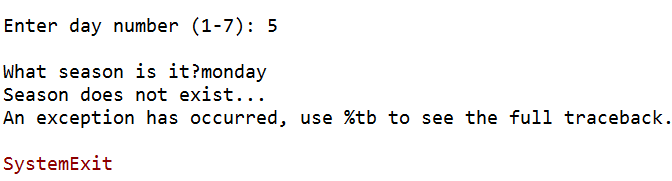
*print(“The day is ”,* ***day****, “ which is day number : ”,* ***day\_num****)*

The second print statement should print out the month and season in a similar way.

1. Write one more if statement which checks if the **season** is summer and if **day\_num** is 6. If it is summer and the variable named **day\_num** holds the value 6 then print out **“Go swimming!”.**
2. Save and run your program. If there aren’t any errors then your output should be similar to the ones below. Run your program a few times, trying out different values for input.







**Rubric**

Upload your source code to Canvas. Here's what we are looking for, when grading your submission. In later lab submissions, there will be additional items that will be graded, which will be described in future lectures.

The .py file must be thoroughly commented. If your code breaks (crashes) because you've been unable to fix a syntax error, then the comments will allow you to receive partial credit.

For this lab, make sure that the following folder is uploaded to Canvas:

*lab2.zip*

And that it contains the following files:

*Cars.py, SeasonsAndDays.py*

|  |  |
| --- | --- |
| **File / task** | **Points** |
| 1. **Cars.py** | 50 |
| **II. SeasonsAndDays.py** | 50 |
| Total | 100 |