CS 111 – Introduction to Computer Science – Fall 2017

Lab Assignment #5

Selection Statements * *(30pts)*

Due Date: at 11:59pm on Saturday, March 10.

The purpose of this lab is to introduce the use of the `if` statement and logic expressions in Python programs. You will be finding and correcting syntax and logic errors and constructing programs using `if-elif-else` statements.

Before getting started with the lab, copy the entire `lab5` folder from the course folder (`H:\Compsci\givens\cs111`) to your `U:\cs111` folder.

Debugging

Finding syntax and logic errors in programs that use selection statements can be more challenging than programs with simple computations.

To begin, open the `lineSlope.py` program from your `lab5` folder in *Wing* and add your name to the file prolog.

This program is supposed to read the discrete coordinates of a line segment from the user and then compute and display the slope of the line. If the slope cannot be computed, however, it is suppose to print a message and stop.

The program contains several errors, both syntactical and logical. You are to find and correct all of the errors in the program so it will execute correctly. When you are finished making the corrections and the program runs correctly, be sure the file is saved.

■

Temperature

Next, you are going to write your own program from scratch that makes use of the `if` statement. You may want to base your program on those provided in the lecture and this lab.

You are to design and implement a simple program that reads a temperature given in degrees Fahrenheit from the user and determines if the temperature is at or below freezing or is above freezing. Thus, after reading the value from the user, as a whole number, your program should print one of two messages:

*Based on the labs of Dr. Rance Necaise

```
It's above freezing.
```

or

```
IT'S FREEZING!!!
```

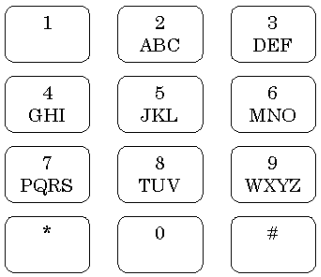
Your program should be written to the following specifications:

- Name your program `freezing.py`
- Include an appropriate file prolog in the source file and provide appropriate comments throughout the program.
- Be sure to use meaningful variable names.
- Use an appropriate prompt when reading the input value and assume the user will enter a valid numerical value.
- Your program must print one of the two messages shown earlier.

After completing your program, be sure to save the source file.
■

Telephone Keys

The modern telephone keypad used in the United States, which is comprised of push button style keys, was standardized in the 1960's. As part of the standard, most of the keys are associated with 3 or 4 letters of the English alphabet. While these were originally used to identify an exchange, today they are used as a way of remembering a telephone number or for entering letters as part of a text string.



 You are to design and implement a program (`telephone.py`) that can be used to convert a letter to its corresponding digit on the telephone. Your program should prompt the user for a letter

```
Enter a letter (must be upper case):
```

and then display the corresponding digit on the telephone keypad. For example, assuming the user enters `P` at the prompt, the program should display

```
The letter P corresponds to 7 on the telephone.
```

If an invalid character is entered, the program should display an error message and stop. For example, if the user enters a `@` at the prompt, your program should display

```
Error!! bad input character: @
```

Your program should be written to the following specifications:

- Include an appropriate file prolog and appropriate comments throughout the program.
- Only upper case letters should be considered valid input.
- You must use the `if-elif-else` version of the `if` statement.

After completing the program, be sure to fully test the program by running the program multiple times and entering various input values, both valid and invalid.
■

ATM Menu

Suppose you are asked to write a simple ATM menu system for a bank. When standing before the ATM, the client is able to either withdraw money or deposit money, as long as the actions are valid. Invalid actions, such as overdrawing the account balance and depositing/withdrawing a negative amount of money, are forbidden.

 You are to write a program named `atm.py` that mimics a simple ATM menu system. The initial ATM screen should look roughly like the following:

```
Welcome to the CS111 Bank
Your current balance is $2100
Please choose either   (D)eposit   or   (W)ithdrawal
```

After choosing an action, either deposit or withdrawal, your program should prompt for and read the amount to be deposited

```
Enter the amount to be deposited $
```

or withdrawn

```
Enter the amount to be withdrawn $
```

The current balance, which should be set to $2100, should then be updated as appropriate and the ending balanced displayed to the terminal:

```
Your current balance is $2300
```

Your program should be written to the following specifications:

- Include an appropriate file prolog and appropriate comments throughout the program.

- The choice of action must be indicated by a single letter, either upper or lower case.
- The dollar amount entered by the user must be positive.
- In the case of a withdrawal, the dollar amount entered must not be greater than the current balance.
- If the user enters invalid input, the program should display an appropriate error message and stop.
- You must use nested `if-else` statements.

After completing the program, be sure to fully test the program. This should include testing various input values for the various user prompts to ensure that your program catches all possible input errors.
■

Finishing Up

When you are finished with the lab, you need to show me that your code runs and correctly computes the solution for each part of the lab. Also, you need to submit the source files for grading. To submit the files, find the lab assignment on Canvas and upload the four files:

- `lineSlope.py`
- `freezing.py`
- `telephone.py`
- `atm.py`

Remember, all of the files must be named exactly as indicated above, with the same case and with no spaces or special characters.