CS 1301

Individual: Homework 4 – Getting to Know your Robot

Due before 11:55pm on Friday September 26th, 2014.

Files to submit: hw4.py

This is an INDIVIDUAL assignment!

Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 1301, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TAs
- Instructor's office hours

Notes:

- Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).
- Do not wait until the last minute to do this assignment in case you run into problems.
- If you find a significant error in the homework assignment, please let a TA know immediately.

Part 1: Tricks and Tunes! (50 points)

You are super excited that your mom let you get a new pet. A puppy? A kitten? No, this is a super cool hybrid anything-you-want-it-to-be creature that just happens to look and behave like a little robot.

You can find a reference to Myro functions that will help you and your little robot to learn how to perform simple movements here: http://calicoproject.org/Calico_Myro.

Your job is to train your little pet robot to perform some tricks. Because your pet is not a normal animal and is a from-the-future being, it must perform combinations of tricks.

> Examples of tricks:
>
> -Moves forward and returns to you (a game similar to fetch?)
>
> -Spins around "chasing its own tail" (you can imagine your creature to have a tail, wings, anything else that gives you an idea for a new trick).
>
> -Moving around the room very slowly (crawling on its belly)
>
> You are allowed (and encouraged) to make your own helper functions that contain individual tricks. Try to include comments that specify what the trick is intended to be.

However, your job is not done yet! It sometimes gets too quiet and you wish your little robot could communicate with you. Get familiarized on how to use the beep() function to create different tones. You are encouraged to write a helper function for these sounds as well. Your task is basically to make your robot come up with a simple melody that it greets you with every morning! This does not include beeping in a single tone. The tune must have between 5 and 10 notes.

Finally, write your robot’s combination of its favorite tricks and melody as a function called **morningRoutine()**, and save it into a file called hw4.py. You will have points taken off if your function and/or file are not named correctly.

Have your morningRoutine function accept in a single integer parameter that specifies how many tricks to do. **(< 1 means do zero tricks, 1,2,3,4 will do the specific number of tricks, and >4 will do all four tricks and the tune!)** Note that this means your program must have **at least four tricks written in it** so that if the user enters a number greater than 4, it will perform four tricks and the tune. You may use the morningRoutine function when writing your greetMenu() function.

Part 2 Write Menu (50 points)

So you’ve succeeded in training your robot and bonding with it through your morning chirrups. Now you want to see how well your robot connects with other people!

Write a menu function called **greetMenu()** that allows the user to give your robot different treats. Depending on the treats given, the robot should respond with different tricks and tunes. Less exciting treats will make the robot do fewer tricks (at least two) and sing a slow and low-pitched melody. More exciting treats will make the robot do more tricks and sound excited with a faster and higher-pitched melody. (Frequency and timing of the notes can be adjusted using the beep() function).

Your menu must offer at least **three different options for treats, and an option to exit the program.** If the user enters an incorrect choice, print the appropriate message and give them the user menu once again.

SAMPLE INPUT:

Here's an example of how greetMenu() might work. The user's input is shown in red.

```
1. Tiny Treats
2. OK Treats
3. Jumbo Treats

0. Exit
```

```
Which option would you like? 1(The robot does a few tricks and
sings a slow and low-pitched tune. The program doesn't show the
menu again until these actions finish a few seconds later.)

1. Tiny Treats
2. OK Treats
3. Jumbo Treats

0. Exit
Which option would you like? 4

I'm sorry, I cannot accept that.
1. Tiny Treats
2. OK Treats
3. Jumbo Treats

0. Exit
Which option would you like? 0
Bye, bye! Have a good day!
(greetMenu() terminates.)
```

Part 3 Turning It In

Once you are done, submit **hw4.py** to TSquare. Remember, if there are serious errors in a part of the homework, you will lose credit for that particular part. However, ***if your file completely fails to run, it will not be graded and will receive an automatic zero.*** Please test your code thoroughly before turning it in!

Rubric

Part 1 Tricks and Tunes (50 points)

Function named correctly (morningRoutine())	**5 points**
Trick routine has at least four tricks (5 points per move)	**20 points**
Robot produces a short tune	**10 points**
Function does correct number of tricks based upon input parameter –	**15 points**

Part 2 Write Menu (50 points)

Function named correctly (greetMenu())	5 points
Displays a menu with options	5 points
Accepts input from the user	10 points
Loops and exits correctly	10 points
Handles cases correctly with conditionals	20 points

You can earn up to 5 points bonus [discretion of the TAs] for extra creativity/general awesomeness, for a possible total of 105/100.