Introductory course in programming (DAT425)

27 October 2020

Examiner: Krasimir Angelov

Note: Solutions must be uploaded to Canvas as a single Python file. It is not necessary to comment on your code, it is more important that the code is clean and easy to read.

All help materials are allowed, but everyone is expected to work alone. Cooperation is not allowed and cheating may lead to suspension from the university.

10 points correspond to grade 3, 17 points correspond to grade 4, and 24 points correspond to grade 5.

Question 1

When taking a loan for buying an apartment or house the following rules apply:

- If the loan is 70% or more of the total property price, 2% of the total price must be paid back to the bank per year.
- If the loan is 50% or more of the total property price, 1% of the total price must be paid back to the bank per year.
- If the loan is less than 50% of the total property price, the buyer doesn't pay back the loan, but he/she must still pay bank interest.

Every bank has a rate in percent which states how much interest must be paid for the loan per year.

Write a function calculate_loan() which when executed calculates the expenses for the loan. It first asks the user for the property price, the size of the loan and the interest rate, and then prints back how much of

the loan must be paid back (amortized) per month, the size of the monthly interest, and finally the total monthly expenses.

Note: the rules in the first paragraph are all stated per year, while you must print the expenses per month, i.e. you need to divide by 12.

Here are two examples for a dialog with the program:

```
>>> calculate_loan()
property price: 1800000
loan:              1000000
interest rate:    1.2

per month
-------------
amortization: 1500.0
interest:        1000.0
total:             2500.0


>>> calculate_loan()
property price: 1500000
loan:              600000
interest rate:    1.2

per month
-------------
amortization: 0.0
interest:        600.0
total:             600.0
```

(7 points)

Question 2

Implement the function:

```
code_words(text, dictionary)
```

Which takes as the first argument an arbitrary text as string, and returns a new version of the text where certain code words are replaced with other words. The second argument is the dictionary for code words.

For example:

```
>>> d = {"happiness": "cake", "homework": "date"}
>>> print(code_words("you have your happiness", d))
you have your cake
>>> print(code_words("I have a homework today", d))
I have a date today
```

All words in the text are separated by spaces. Words that are not in the dictionary should be left unchanged.

(10 points)

Question 3

Write a class called Robot which models the state of a robot which can move over a surface. It can follow three commands: turnLeft, turnRight and forward N. The first two commands tell the turtle to turn to the left/right while the last tells it to go forward with N number of steps.

The class must maintain the current coordinates of the robot and its direction. The direction is one of the strings: "NORTH", "SOUTH", "EAST", "WEST", and indicates which world direction the robot is currently facing. The class must also have the following methods:

- __init__(self): must initialize the state of the object and set the current coordinates to be x=0 and y=0. The initial direction must be “NORTH”.
- turnLeft(self): tells the robot to turn to the left
- forward(self, n): tells the robot to make n number of steps forwards.

Command turnRight would be very similar to turnLeft, so we skip it for the exam.

Note that, at all times, the robot is facing a particular direction, and the behaviour of each of the commands depends on the current direction. For convenience the effect of each command depending on the current direction is shown in the table:

	turnLeft	turnRight	forward
NORTH	face WEST	face EAST	increase Y
SOUTH	face EAST	face WEST	decrease Y
EAST	face NORTH	face SOUTH	increase X
WEST	face SOUTH	face NORTH	decrease X

For example, if we have:

```
r = Robot()
r.forward(10)
```

then the final position will be x=0, y=10 because initially the robot was at x=0, y=0 and it is looking towards NORTH.

Similarly if we have:

```
r = Robot()
r.turnLeft()
r.forward(10)
```

then the robot will first change its direction to face WEST and then it will change its x coordinate. The final position will be x=-10, y=0.

(13 points)