30.10.2023

CENG 113 – Programming Basics

Assignment 1

Due: 10.11.2023 - 23:59

Topics

- Input, Processing & Output
- Decision Structures & Boolean Logic
- Repetition Structures

When designing the program, you are expected to use only the topics covered in the course so far. We expect you to avoid using topics that have not yet been covered. (E.g., user defined functions, lists)

Assignment Description

In this assignment, you are expected to develop a simplified version of the famous card game twenty-one, played against the computer, using the Python programming language. In the program to be developed, the computer plays the role of dealer; The user is in the role of the player.

The traditional twenty-one game is a game played with playing cards. The value of each card in the game is equal to the number on them. The values of jack, queen and king are 10. Another exception is that the value of Ace (i.e., 1) can be used as 1 or 11, depending on the cards in the player's hand.

The aim of the game is for the total value of the cards in the hand to be more than the opponent's but not to exceed 21.

Flow of the game to be designed:

At the beginning of the game, the dealer (computer) and the player (user) each have 1000 points. When each round starts, the player is asked the amount of points he will sacrifice/gain. Then the dealer deals 2 cards each to itself and the player. At this stage, the player can only see one of the dealer's cards. Then, depending on the score and cards, the player is given two options:

- If the total value of the cards in the player's hand is less than 21 and their current point is sufficient, they are asked if they want to double the points of the round by drawing just one more card. If the player chooses this option, they receive one more card, their turn ends and the points for the round is doubled. The dealer is free to take as many cards as it wants during his turn.
- If the conditions are not met or the player does not choose the first option, the player is asked if they want a new card. The card drawing process continues as long as the player requests a card. If the player exceeds 21 with the cards they receive, they lose the round. When the player stops asking for new cards, their turn for that round is completed.

When the player completes their turn, the cards in the dealer's hand are shown to the player. The dealer's goal is to match or beat the total value of the player's cards. The dealer starts drawing cards one by one and if it can exceed the value of the player's total cards without exceeding 21, dealer wins the round. If the total values of the player and dealer's hands are equal, the dealer stops drawing cards and the round ends in a draw. If the dealer goes over 21, dealer loses the round.

At the end of each round, the winning side's total points increases by the amount determined at the beginning of the round; The losing side's points decreases by the same amount. If the round ends in a draw, there will be no change in the points.

When the total score of either the player or the dealer reaches 0, the game is over and the winner of the game is determined.

Keypoints:

If there is an Ace (i.e., 1) among the cards in hand, the program should automatically determine the value of Ace according to the cards in hand after each card draw. When the value of Ace is 11, if the total value of the player's cards does not exceed 21, the value should stay 11. Otherwise, it should be used as 1.

Functions that may be useful:

The **“randint”** function in Python can be used to draw random cards. After adding the **“from random import randint”** command to the beginning of the program, the **“card = min(10, randint(1, 13))”** statement can be used to generate a random card. (Since the probability of encountering a card with a value of 10 is higher than others)

Requirements

Please read the assignment description and sample program output carefully and design your program in the required flow. **It is expected that the submitted program will work in accordance with the described flow and have the specified restrictions and features.**

Submission Rules

- Assignments must be done and submitted individually. Students found to have cheated will automatically receive 0 points.
- You must submit your assignment by the specified deadline.
- The submitted file must contain your source code and **code explanations (Comments)**.
- The name of the submitted file must be your student number. **(E.g., 123456.py)**
- Only Python files with .py and .ipynb extensions will be evaluated. **Files with .txt, .pdf, .jpeg etc. extensions will not be accepted.**
- **Please submit your file without compressing it.**

Sample Output in Expected Format

```
Welcome to the game!
##############################
Round 1
Current User Points:  1000
Current Dealer Points:  1000
##############################

Please place a maximum of 1000 bet:
```

```
200
Dealer has: ? - 1
User has: 2 - 6 (Total: 8)

Do you want to draw just one more card and double the bet? (y,n)
n
Do you want another card? (y,n)
y
New Card:  10

User has: 2 - 6 - 10 (Total: 18)
Do you want another card? (y,n)
n

Dealer has: 5 - 1 (Total: 16)

Dealer draws a new card: 7
Dealer has: 5 – 1 – 7 (Total: 13)

Dealer draws a new card: 1
Dealer has: 5 – 1 – 7 – 1 (Total: 14)

Dealer draws a new card: 9
Dealer has: 5 – 1 – 7 – 1 - 9 (Total: 23)

User wins the round.

Press Enter to continue...

##############################
Round 2
Current User Points:  1200
Current Dealer Points:  800
##############################

Please place a maximum of 800 bet:
…
```