



3 Classes exist in this program.

Player, CardDeck, Blackjack

Player class is for holding information about the player since they will be multiple player in the game so it's make sense to make a class to hold those information individually.

CardDeck class is for holding the method related to getting the card and store a list of card.

Blackjack class is for containing all the method to operate the game and include all the attributes need for operating the game.

Method in Player class:

__init__ : Set up attributes for player object

Method in CardDeck class:

__init__(self): Set up attributes for deck object

shuffle(self): Shuffle the deck draw_cards(self,n):

Return n card drawn from deck.

Method in Blackjack class:

__init__(self): Set up attributes for blackjack game object

start(self): Start the game(Draw card and adjust value for all players including computer)

status_update(player): Check whether current status of a player is legit or not.

value_cal(player): Calculate the value in the hand of a player.

ace_cal(player , num_ace): Calculate whether the ace in the hand should be 1 or 11.

adjust_player_hand(self, player): Adjust the hand of player(Draw more) according to the status of the player.

display_player_hand(player) : Show the hand of the player to the screen.

decision(self): Check who is winning and print out result + plus the point to the winner.

Sample:

The image displays two screenshots of a PyCharm IDE window, showing the development of a Python program titled "PLAY_BLACKJACK (2)".

Top Screenshot:

- Project Structure:** The left sidebar shows a project named "Final Project" with subfolders "task1", "task2", and "task3". The "task3" folder is selected, containing files like "QuizOOP", "6310546058_Poomtum_ex8.zip", "README.md", "Game", "Hacktober", "hacktoberfest", "HTML_CSS", and "inclass_code (1)".
- Code Editor:** The main editor shows the "play_blackjack.py" file. The code defines a "def menu()" function that uses a "while True:" loop to prompt the user for menu choices. Choices include "1. Play game", "2. Add player's budget", "3. View player's profile", "4. View stat", and "5. Quit program". It also prompts for the number of players and initializes a "global game" object.
- Run Console:** The bottom panel shows the output of the program. It displays the menu options and the user's input: "Enter menu choice: 1", "How many players? 4", "Bessie", "7 Jack", and "stay or not:".

Bottom Screenshot:

- Code Editor:** The code editor shows the continuation of the "play_blackjack.py" file. It includes logic for adding a player's budget and viewing player statistics. The "def view_player(game)" function prompts for the player's name and iterates through the game's players to display their budget, wins, and losses.
- Run Console:** The bottom panel shows the updated output. It displays the menu options and the user's input: "Enter menu choice: 2", "Enter player's name: Martha", "Enter added budget: 500", "500 Added", "Martha, F, Midwest", and "Budget = 1900".

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Programming - play_blackjack.py
Programming > compro1 > Final Project > task3 > play_blackjack.py
Project
  6310546058_Poomtum_inclass9
  Final Project
    inclass_final_project
    task1
    task2
    task3
    QuizOOP
    6310546058_Poomtum_ex8.zip
    README.md
  Game
  Hacktober
  hacktoberfest
  HTML_CSS
  inclass_code (1)
Run: play_blackjack (2) x
  5.Quit program
  Enter menu choice: 3
  Enter player's name: Peter
  Peter, M, Midwest
  Budget = 1600
  Number of wins = 154
  Number of losses = 70
  >>>Main menu
  1.Play game
  2.Add player's budget
  3.View player's profile

68 player = input("Enter player's name: ")
69 budget = int(input("Enter added budget: "))
70 for i in game.players:
71     if i.player_name == player:
72         i.budget += budget
73         print(f"{budget} Added")
74         print(
75             f"{i.player_name}, {i.gender}, {i.region}\nBudget = {i.budget}\nNumber of wins = {i.num_wins}\nNumber of losses = {i.num_losses}")
76
77 def view_player(game):
78     player = input("Enter player's name: ")
79     for i in game.players:
80         if i.player_name == player:
81             return f"{i.player_name}, {i.gender}, {i.region}\nBudget = {i.budget}\nNumber of wins = {i.num_wins}\nNumber of losses = {i.num_losses}"
82
83 menu()
while True
```

PyCharm 2020.2.4 available // Update... (today 11:37 AM)

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Programming - play_blackjack.py
Programming > compro1 > Final Project > task3 > play_blackjack.py
Project
  6310546058_Poomtum_inclass9
  Final Project
    inclass_final_project
    task1
    task2
    task3
    QuizOOP
    6310546058_Poomtum_ex8.zip
    README.md
  Game
  Hacktober
  hacktoberfest
  HTML_CSS
  inclass_code (1)
Run: play_blackjack (2) x
  5.Quit program
  Enter menu choice: 4
  1.Bertha, F, South
  Budget = 4000
  Number of wins = 200
  Number of losses = 64
  2.Mattie, F, West
  Budget = 2300
  Number of wins = 199
  Number of losses = 91
  3.Mary, F, West

68 player = input("Enter player's name: ")
69 budget = int(input("Enter added budget: "))
70 for i in game.players:
71     if i.player_name == player:
72         i.budget += budget
73         print(f"{budget} Added")
74         print(
75             f"{i.player_name}, {i.gender}, {i.region}\nBudget = {i.budget}\nNumber of wins = {i.num_wins}\nNumber of losses = {i.num_losses}")
76
77 def view_player(game):
78     player = input("Enter player's name: ")
79     for i in game.players:
80         if i.player_name == player:
81             return f"{i.player_name}, {i.gender}, {i.region}\nBudget = {i.budget}\nNumber of wins = {i.num_wins}\nNumber of losses = {i.num_losses}"
82
83 menu()
while True
```

PyCharm 2020.2.4 available // Update... (today 11:37 AM)

Conclusion:

Most of the things work except for one thing. The game can't find the winner so they're some numbers of problems happen because of that.