

Topic 1 – Modules and Packages - Assignment

Instructions:

There are two exercises in this assignment. Attempt both questions. Sample solutions are made available for both exercises. Please post any queries about the exercises in the Topic 1 forum on the eCollege course.

Exercise 1: Card Game: Blackjack / 21

Create a program to create the card game, Blackjack / 21.

A player is dealt 3 random cards (from a pack of 52) with values ranging from 1-13. The program should generate three random cards. The program should ensure that duplicate cards are not dealt in the same round. For example, the *Six of Clubs* should not appear twice in the same round.

Aces are low in value (1) and Kings are high in value (13). Jokers aren't played.

The player wins / beats the house if the sum of the 3 cards dealt is less than or equal to 21. Otherwise, the house wins.

Two demo runs of the program are shown below.

```
*****Welcome to BlackJack - the game of 21!*****

The computer will deal three random cards.
To win, the total value of the three cards dealt must be less than or equal to 21. Otherwise, house wins!
Start Game...

SEVEN of SPADES Card Value: (7)
TEN of DIAMONDS Card Value: (10)
FIVE of CLUBS Card Value: (5)

House wins! 22 is greater than 21

Better luck next time! We value your custom. Thanks for playing.
```

Fig. 1.1: The house wins this time in the game of Blackjack / 21.

```
*****Welome to BlackJack - the game of 21!*****

The computer will deal three random cards.
To win, the total value of the three cards dealt must be less than or equal to 21. Otherwise, house wins!
Start Game...

TWO of HEARTS Card Value: (2)
FIVE of SPADES Card Value: (5)
SIX of SPADES Card Value: (6)

Patron wins! 13 is less than 21
```

Fig. 1.2: Victory for the player in the game of Blackjack / 21.

Exercise Requirements:

- Variables and functions should be named using the snake_case naming convention.

Exercise 2:

World Cup 2026 (Canada, Mexico and the United States)

You are working as a Junior member of a Software Development team.

In your role, you will be working with senior programmers in creating a new football simulation game for the World Cup 2026. The game will simulate a real match day experience at the World Cup including opening a stadium, selling tickets, offering hospitality and of course, watching the football game.



Your role as a junior member of the team is to start creating the various modules that are required and to put them into a package.

Create the following directory hierarchy.

Q2

```
-py
  -packages
    -world_cup
      -hospitality
        -corporate.py
        -standard.py
      -media
        -internet.py
        -radio.py
        -tv.py
      -stadia
        -stadium_manager.py
      -ticketing
        -corporate.py
        -standard.py
    -__init__.py
  -progs
    -main.py
```

For the purposes of the exercise, the functionality outlined in the modules will be kept simple.

Folder	Module	Function
hospitality	corporate.py	<i>def corporate(): return "Launch Corporate Hospitality"</i>

hospitality	standard.py	<i>def standard(): return "Launch Standard Hospitality"</i>
-------------	-------------	---

Folder	Module	Function
media	internet.py	<i>def view_game(): return "View game on Internet"</i>
media	radio.py	<i>def hear_game(): return "Hear game on Radio"</i>
media	tv.py	<i>def view_game(): return "View game on TV"</i>

Continued Overleaf

Folder	Module	Functions
stadia	stadium_manager.py	<i>def open(): return "Open Stadium"</i> <i>def close(): return "Close Stadium"</i>

Folder	Module	Function
ticketing	corporate.py	<i>def corporate(): return "Launch Corporate Ticketing"</i>
ticketing	standard.py	<i>def standard(): return "Launch Standard Ticketing"</i>

Finally, in the *main.py* file in the progs folder (see image of directory structure outlined earlier in the question), test the functionality outlined in the various modules. Import all the various modules and call the functions. A demo of *main.py* executing is shown below.

```
Open Stadium
Launch Corporate Ticketing
Launch Standard Ticketing
Launch Corporate Hospitality
Launch Standard Hospitality
View game on Internet
Hear game on Radio
View game on TV
Close Stadium
```

Continued Overleaf

Exercise Requirements:

- Variables and functions should be named using the snake_case naming convention.

END OF ASSIGNMENT

