Quizshow.cpp

User manual

Program: quizshow.cpp

- This program implements the quiz show strategy for 10,000 times and prints the number and percentage of time the player can win if he/ she changes his/ her choice and also when he/ she does not changes his/ her choice.
- To run the program double click on thequizshow.exe.
- A console window will appear showing the number of times a person won if he/she changed his/ her choice, number of times he/ she won if the choice was not changed and also the percentage win for both cases.
- It also pints if the person should consider changing his/ her choice or not.
- After the output is displayed, press any key to exit the program and close the console.

System Manual

Program: quizshow.cpp

- This program implements the quiz show strategy for 10,000 times and prints the number and percentage of time the player can win if he/ she changes his/ her choice.
- It contains a main function and three user-defined functions : setupDoors, pickDoorChoices and checkstrategy.

Main function :

```
int main()
```

- Three variable of character data type named door1, door2 and door3 are declared to store the character G or C.
- "G" stands for goat and "C" stands for Car.
- Two variable of integer data type named doorPlayer and doorMonty are declared.
  doorPlayer stores the door that the player choose.
  doorMonty stores the door that Monty chooses.
- Three variables of integer data type named strategy1, strategy2 and total are declared and initialized.
  strategy1 is initialized to 0 and its value increases when the car is not behind the door that player chose and should consider changing his/ her choice.
  strategy2 is also initialized to 0 and its value increases when the car is behind the door that player chose and should not change his/ her choice.
  total is initialized to 10000. It stores the number of trial that need to be carried out before deciding whether the player should change his/ her choice or not.
- Two variable of float data type called strategy1_percent and strategy2_percent are declared. These two variables stores the percent of strategy1 and percent of strategy2 out of total trial carried out.
- This function then calls setupDoors, pickDoorChoices and checkstrategy for number stored in total variable times.
- Then percent of number of times strategy1 won and percentage of number of time strategy2 won is calculated using the following formula :

```
strategy1_percent = ((float) strategy1 / (float) total)*100;
strategy2_percent = ((float) strategy2 / (float)total)*100;
```

- At the end outputs are displayed.

setupDoors :

```
void setupDoors(char &door1, char &door2, char &door3))
```

- This is a user defined function that randomly assign G or C character to each character variable ddor1, door2 and door3.
- One of the door is assigned value of C and the other two are assigned value of G.
- First this function generates a random integer from 1 to 3 and store it in a integer variable named "a".
- If the value of "a" is 1, door1 is assigned the value "C" and the rest two doors are assigned value "G".
- If the value of "a" is 2, door2 is assigned the value "C" and the rest two doors are assigned value "G".
- If the value of "a" is 3, door3 is assigned the value "C" and the rest two doors are assigned value "G".
- The variables door1, door2 and door3 are called by reference so any changes made to these variables are reflected back.
- *This function is called in main function only.*

pickDoorChoices:

```
void pickDoorChoices(char door1, char door2, char door3, int &doorPlayer, int
&doorMonty)
```

- This function randomly makes the player to choose one of the door(assigns variable doorPlayer a random door)
- It then makes Monty choose a door that does not have car in it.
- variables doorPlayer and doorMonty are the value passed by reference to this variable.
- *This function is called only in main function and does not return any value.*

checkstrategy:

```
void checkstrategy(char door1, char door2, char door3, int&doorPlayer, int&strategy2,
int&strategy1)
```

- The function is called from the main function.
- It check which of the strategy works for the player to win.
- To do this, it checks the value of doorPlayer. It then checks the stored in that door.
  For example, if the value of doorPlayer is 1, this function checks the value in door1.
- If the value stored in that door is "C", strategy 2 is incremented. That indicates that the player should not change his/ her choice.
- If the value stored in that door in not "C", then the value of strategy1 is incremented. That indicates that the player can win if he/ she changes his/ her choice.