**Assignment 05**

**REQUIREMENTS**

Start up the IDE and create a new Console Application project named ProgrammingAssignment5.

Your program must do the following (all in the Main method):

* Declare a variable for a random number generator and create a new Random object for that variable
* Declare other variables as necessary to keep track of each player’s roll, each player’s number of wins, and whether or not we should play another game of War
* Print a “welcome” message to the user telling them that the program will play games of War
* While the player wants to play another game
  + Set both player win counts to 0
  + Loop for 21 battles
    - Roll the die for player 1
    - Roll the die for player 2
    - While the rolls have the same value (a tie)
    - Print “WAR” and the die values for the two players
    - Roll the die for player 1
    - Roll the die for player 2
    - Print the die values for the two players
    - Print which player won and increment that player’s win count
  + Print out which player won more battles
  + Prompt for and get whether the player wants to play another game

**ASSUMPTIONS**

* Users may enter any character when asked to play again. Accept only ‘n’ or ‘N’ to end the program. Any other entry plays another game of War.
* If you develop your solution to this problem using the appropriate programming constructs, you should have the following constructs in your application:
  + for loop: executes the 21 battles for each war
  + while loops:
    - rolls the die again whenever there is a tie [there can be multiple wars in a row!]
    - keeps playing games of war
  + if statements:
    - determines battle winner
    - determines war winner

**EXAMPLE OUTPUT**

Here’s some example output. You don’t have to match this exactly, but your output format should look pretty close to this (your player rolls will obviously be different). You definitely need to make sure the columns are lined up for the player rolls and winner messages.

Welcome to War!

BATTLE: P1:6    P2:8    P2 Wins!

BATTLE: P1:6    P2:10   P2 Wins!

BATTLE: P1:8 P2:3 P1 Wins!

BATTLE: P1:5 P2:3 P1 Wins!

BATTLE: P1:13 P2:6 P1 Wins!

BATTLE: P1:11 P2:3 P1 Wins!

BATTLE: P1:8 P2:9 P2 Wins!

BATTLE: P1:5 P2:4 P1 Wins!

BATTLE: P1:9 P2:5 P1 Wins!

BATTLE: P1:2 P2:10 P2 Wins!

BATTLE: P1:4 P2:5 P2 Wins!

BATTLE: P1:6 P2:3 P1 Wins!

BATTLE: P1:11 P2:12 P2 Wins!

BATTLE: P1:12 P2:1 P1 Wins!

BATTLE: P1:5 P2:2 P1 Wins!

BATTLE: P1:6 P2:11 P2 Wins!

BATTLE: P1:9 P2:13 P2 Wins!

BATTLE: P1:10 P2:6 P1 Wins!

BATTLE: P1:8 P2:7 P1 Wins!

BATTLE: P1:7 P2:8 P2 Wins!

WAR! P1:4 P2:4

BATTLE: P1:1 P2:13 P2 Wins!

P1 is the overall Winner with 11 battles!

Do you want to play again (y/n)? Y

BATTLE: P1:10 P2:5 P1 Wins!

BATTLE: P1:7 P2:5 P1 Wins!

BATTLE: P1:1 P2:9 P2 Wins!

BATTLE: P1:12 P2:4 P1 Wins!

BATTLE: P1:5 P2:9 P2 Wins!

BATTLE: P1:3 P2:1 P1 Wins!

BATTLE: P1:2 P2:3 P2 Wins!

BATTLE: P1:12 P2:9 P1 Wins!

WAR! P1:5 P2:5

BATTLE: P1:11 P2:12 P2 Wins!

WAR! P1:1 P2:1

WAR! P1:8 P2:8

WAR! P1:12 P2:12

BATTLE: P1:9 P2:2 P1 Wins!

BATTLE: P1:11 P2:8 P1 Wins!

BATTLE: P1:6 P2:13 P2 Wins!

BATTLE: P1:4 P2:7 P2 Wins!

WAR! P1:7 P2:7

BATTLE: P1:6 P2:7 P2 Wins!

BATTLE: P1:13 P2:10 P1 Wins!

BATTLE: P1:4 P2:2

P1 Wins!

BATTLE: P1:7 P2:13 P2 Wins!

BATTLE: P1:11 P2:9 P1 Wins!

WAR! P1:5 P2:5

BATTLE: P1:8 P2:12 P2 Wins!

BATTLE: P1:3 P2:13 P2 Wins!

BATTLE: P1:7 P2:3 P1 Wins!

P1 is the overall Winner with 11 battles!

Do you want to play again (y/n)? n

Press any key to continue . . .

**HELPFUL HINTS**

Write 5 or fewer lines of code, save, compile, test, repeat! Don’t try to write a whole block of code at once – implement one small chunk at a time.

You'll probably want to use the \t escape sequence (to add a tab) when you output each row above to line up the columns properly