Guessing Game & Functions

Due: Tuesday, April 22 - class time!!
*** Turn in a print-out of your code. *** You will be given instructions in class as to how you will turn in your executable code.

You are to write a program called The Guessing Game. (This is like the High-Low game on “Price is Right” if you are familiar with that.)

In this game the computer will generate a random number between 1 and 100 for the player to guess. On each guess, the program is to tell the player if the guess is too high, too low, or correct. The game will only allow 10 guesses. On the 10th guess the player is told the answer. The program must also ask the player if he/she wants to play again.

You must write the following functions as part of your program:

- **int Get_Number ( )**: returns an integer between 1 and 100, inclusive, to be guessed. Must use the built-in function rand. (See pages 128-130.)
  *Hint*: While you are working on your program, print the random number as it is generated so that you can test your guesses to be sure it is working.

- **void Print_Instructions ( )**: This function announces the game and gives the instructions.

- **void End_Game ( )**: This function is called when the user types “n” because they do not wish to play again. It prints “Thank you for playing” and the extra credit parts below (if you do them). If you do the extra credit parts, you will need to add parameters to the parameter list.

The main program drives the game and calls the appropriate functions. It does the following:

*if the guess is correct, print out the winning message, the number of guesses they used.
*if the guess is wrong, print out “too high” or “too low”, tell how many guesses they have remaining, ask for the next guess.
*if the guess is wrong but the player has used the 10 guesses, print the losing message.

This is an overview of the program without the part asking if they want to play again. Get this running first then add the repetition for multiple games. Make sure that it works if they type either an upper case or a lower case letter. (Remember, this is called incremental development.)

**void main ( )**  // below is an outline for main
declare & initialize necessary variables
call Get_Number to obtain the number to be guessed
Call Print Instructions to obtain players first guess
While Loop  // loop until 10 guesses or until number is guessed
   count guesses
Call End-Game
The following variables are to be used in main as described:

secret: represents the number to be guessed
guess: holds the current input (guess) from the player
num_guesses: counts the number of guesses the player has made.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Your output to the screen should look similar to the following:

Welcome to the guessing game written by YOUR NAME HERE.
I will pick a number between 1 and 100 – you try to guess it.
You will have ten chances to guess. I will give you hints as you go to help you.

Are you ready to play? Please type your first guess and hit enter.

Guess 1:  34
Sorry that is too high, try again.
Guess 2:  25
Sorry that is too low, try again.
Guess 3:  30
Right! You are a winner in 3 guesses!

Would you like to play again? (Type y or n.)

Extra Credit Options:
Thanks for playing. You played 7 games. // insert correct number.
You can receive an additional 5 points if your output tells how many games were played.
Thanks for playing. You played 5 games. You won 3 and lost 2. // insert correct number of games won & lost. You can receive another 5 points if your closing output tells the number of games won and lost.

Extra Credit is only awarded to programs turned in on time.
Start Today - this isn’t easy!!!

*** BE SURE TO THOROUGHLY TEST YOUR PROGRAM - I WILL!!!***