1. Write a console-based program that prompts the user for an hourly pay rate. If the value entered is less than $7.50, display an error message
2. Write a console-based program that prompts a user for an hourly pay rate. If the value entered is less than $7.50 or greater than $49.99, display an error message; otherwise, display a message indicating that the rate is okay.
3. If money is left in a particular bank for more than 5 years, the interest rate given by the bank is 7.5%, else the interest rate is 5.4%. Write a program that prompt the user for the number of years that the money was left in the bank and display the appropriate interest rate depending on the value input. How many runs should you make to very that it works correctly?
4. Write a console-based program that prompts a user for an hourly pay rate. If the user enters values less than $7.50 or greater than $49.99, prompt the user again. If the user enters an invalid value again, display an appropriate error message. If the user enters a valid value on either the first or second attempt, display the pay rate as well as the weekly rate, which is calculated as 40 times the hourly rate.
5. Write a console-based program that accepts a user’s message and determines whether it is short enough for a social networking service that does not accept messages of more than 140 characters. [Hint: if the user’s input is assigned to the variable message then message.Length with give the number of characters in the input]
6. Write a console-based program for a college’s admissions office. The user enters a numeric high school grade point average (for example, 3.2) and an admission test score. Display the message “Accept” if the student meets either of the following requirements:

A grade point average of 3.0 or higher and an admission test score of at least 60

A grade point average of less than 3.0 and an admission test score of at least 80

If the student does not meet either of the qualification criteria, display “Reject”.

1. Write a console-based program that prompts the user for an hourly pay rate and hours worked. Compute gross pay (hours times pay rate), withholding tax, and net pay (gross pay minus withholding tax). Withholding tax is computed as a percentage of gross pay based on the following:

|  |  |
| --- | --- |
| **Gross Pay** | **Withholding Percentage** |
| Up to and including 300.00 | 10% |
| More than300.00 | 12% |

1. Write a console-based program for a lawn-mowing service. The lawn-mowing season lasts 20 weeks. The weekly fee for mowing a lot under 400 square feet is $25. The fee for a lot that is 400 square feet or more, but under 600 square feet, is $35 per week. The fee for a lot that is 600 square feet or over is $50 per week. Prompt the user for the length and width of a lawn, and then display the weekly mowing fee, as well as the total fee for the 20-week season.
2. To the Lawn application you created in the above question, add a prompt that asks the user whether the customer wants to pay (1) once, (2) twice, or (3) 20 times per season. If the user enters 1 for once, the fee for the season is simply the seasonal total. If the customer requests two payments, each payment is half the seasonal fee plus a $5 service charge. If the user requests 20 separate payments, add a $3 service charge per week. Display the number of payments the customer must make, each payment amount, and the total for the season.
3. Write a console-based application that asks a user to enter an IQ score. If the score is a number less than 0 or greater than 200, issue an error message; otherwise, issue an “above average”, “average”, or “below average” message for scores over, at, or under 100, respectively.
4. You can create a random number that is at least min but less than max using the following statements:

Random ranNumberGenerator = new Random();

int randomNumber;

randomNumber = ranNumberGenerator.Next(min, max);

Write a console-based program that generates a random number between 1 and 10. (In other words, min is 1 and max is 11.) Ask a user to guess the random number, then display the random number and a message indicating whether the user’s guess was too high, too low, or correct.

1. In the game Rock Paper Scissors, two players simultaneously choose one of three options: rock, paper, or scissors. If both players choose the same option, then the result is a tie. However, if they choose differently, the winner is determined as follows:

Rock beats scissors, because a rock can break a pair of scissors.

Scissors beats paper, because scissors can cut paper.

Paper beats rock, because a piece of paper can cover a rock.

1. Create a console-based game in which the computer randomly chooses rock, paper, or scissors. Let the user enter a character, ‘r’, ‘p’, or ‘s’, each representing one of the three choices. Then, determine the winner.
2. Create a console-based lottery game application. Generate three random numbers, each between 1 and 4. Allow the user to guess three numbers. Compare each of the user’s guesses to the three random numbers and display a message that includes the user’s guess, the randomly determined three-digit number, and the amount of money the user has won as follows:

|  |  |
| --- | --- |
| **Matching Numbers** | **Award ($)** |
| Any one matching | 10 |
| Two matching | 100 |
| Three matching, not in order | 1,000 |
| Three matching in exact order | 10,000 |
| No matches | 0 |

Make certain that your application accommodates repeating digits. For example, if a user guesses 1, 2, and 3, and the randomly generated digits are 1, 1, and 1, do not give the user credit for three correct guesses—just one.