3 questions, first two questions are 30 points each, the third question is 40 points, a printed copy is due at the start of the class on the specified date.

Q1

There are several techniques for implementing the sqrt method in the Math class. One such technique is known as the Babylonian method. It approximates the square root of a number, n, by repeatedly performing the calculation using the following formula:

nextGuess = (lastGuess + n / lastGuess) / 2

When nextGuess and lastGuess are almost identical, nextGuess is the approximated square root. The initial guess can be any positive value (e.g., 1). This value will be the starting value for lastGuess. If the difference between nextGuess and lastGuess is less than a very small number, such as 0.0001, you can claim that nextGuess is the approximated square root of n. If not, nextGuess becomes lastGuess and the approximation process continues.

Implement the following method in your program that returns the square root of n:

public static double sqrt(double n)

You will ask the user to enter a number, n, in the main method and pass that number to the method above.

Q2

A palindromic prime is a prime number and also palindromic. For example, 131 is a prime and also a palindromic prime, as are 313 and 757. Write a program that displays the palindromic prime numbers less than 1000. Display 5 numbers per line, separated by exactly one space, as follows:

2 3 5 7 11

101 131 151 181 191

313 353 373 383 727

757 787 797 919 929

Implement the following two methods that are called from the main method. Please do not create any other methods besides these two and the main method.

public static boolean isPrime(int num)

public static boolean isPalindrome(int num)

Q3

Craps is a popular dice game played in casinos. Write a program to play a variation of the game, as follows: Roll two dice. Each die has six faces representing values 1, 2, ..., and 6, respectively. Check the sum of the two dice. If the sum is 2, 3, or 12 (called craps), you lose; if the sum is 7 or 11 (called natural), you win; if the sum is another value (i.e., 4, 5, 6, 8, 9, or 10), a point is established. Continue to roll the dice until either a 7 or the same point value is rolled. If 7 is rolled, you lose. Otherwise, you win.

Your program acts as a single player.

Besides the main method, your program will have one additional method:

public static int getDice()

This method will generate a number between 1 and 6 and return it back to the method call in main. This getDice() method will not print anything. All the printing is done in the main method.

Here are some sample runs.

You rolled 5 + 6 = 11

You win

You rolled 1 + 2 = 3

You lose

You rolled 3 + 2 = 5

point is 5

You rolled 2 + 5 = 7

You lose

You rolled 3 + 1 = 4

point is 4

You rolled 5 + 6 = 11

You rolled 1 + 3 = 4

You win