(Display an integer reversed) Write a method with the following header to display an integer in reverse order:

public static void reverse (int number)

For example, reverse (3456) displays 6543. Write a test program that prompts the user to enter an integer and displays its reversal.

ANS:

```
package displayanintegerreversed;

import java.util.*;
public class DisplayAnIntegerReversed {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a intiger that u want to dispaly reversed: ");
        int number = input.nextInt();
        reverse(number);
    }

    public static void reverse(int number){
        int b;
        for(int a=number;a!= 0;a/=10){
            b=a;
            b%=10;
         System.out.print(b);
        }
    }
}
```

6.8 (Conversions between Celsius and Fahrenheit) Write a class that contains the following two methods:

```
/** Convert from Celsius to Fahrenheit */

public static double celsiusToFahrenheit(double celsius)

/** Convert from Fahrenheit to Celsius */

public static double fahrenheitToCelsius(double fahrenheit)
```

The formula for the conversion is:

fahrenheit = (9.0 / 5) * celsius + 32 celsius = (5.0 / 9) * (fahrenheit – 32)

Write a test program that invokes these methods to display the following tables:

```
Celsius      Fahrenheit    |    Fahrenheit    Celsius
40.0         104.0         |    120.0         48.89
39.0         102.2         |    110.0         43.33
 ...
32.0         89.6          |    40.0          4.44
31.0         87.8          |    30.0          -1.1
```

ANS:

```
package conversions.between.celsius.and.fahrenheit;
public class ConversionsBetweenCelsiusAndFahrenheit {
   public static void main(String[] args) {
      double celsius = 40;
      double fahrenheit = 120;
      double a,b;
      System.out.print("Celsius\t\tFahrenheit\t|\tFahrenheit\tCelsius\n");
System.out.println("_____________________________________________________________________
");
      for(int c=1;c<11;c++){
      a = celsiusToFahrenheit(celsius);
      b = fahrenheitToCelsius(fahrenheit);
      System.out.print(celsius+"\t\t");
      System.out.printf("%-1.2f\t\t|\t",a);
      System.out.print(fahrenheit+"\t\t");
      System.out.printf("%-1.2f\n",b);
      celsius--;
      fahrenheit-=10;
      }
   }
   public static double celsiusToFahrenheit(double celsius){
      double fahrenheit = (9.0 / 5) * celsius + 32;
      return fahrenheit;
   }
   public static double fahrenheitToCelsius(double fahrenheit){
      double celsius =(5.0 / 9)*(fahrenheit - 32);
      return celsius;
   }
}
```

6.10 (Use the **isPrime** Method) Listing 6.7, PrimeNumberMethod.java, provides the **isPrime(int number)** method for testing whether a number is prime. Use this method to find the number of prime numbers less than **10000**.

ANS:

```
package usetheisprimemethod;

public class UseTheIsPrimeMethod {

  public static void main(String[] args) {
  System.out.println("The prime numbers less than 10000 are \n");
    printPrimeNumbers(10000);
    }

  public static void printPrimeNumbers(int numberOfPrimes) {
    final int NUMBER_OF_PRIMES_PER_LINE = 10;
    int count = 0;
    int number = 2;

    while (count < numberOfPrimes) {
    if (isPrime(number)) {
    count++;
    if (count % NUMBER_OF_PRIMES_PER_LINE == 0) {
    System.out.printf("%-5s\n", number);
    }
    else
    System.out.printf("%-5s", number);
    }
    number++;
    if(number>=10000)break;
    }
    }

  public static boolean isPrime(int number) {
    for (int divisor = 2; divisor <= number / 2; divisor++) {
    if (number % divisor == 0) {
    return false;
    }
    }
    return true;
  }

}
```

**6.18 (Check password) Some websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:
■ A password must have at least eight characters.
■ A password consists of only letters and digits.
■ A password must contain at least two digits.
Write a program that prompts the user to enter a password and displays **Valid Password** if the rules are followed or **Invalid Password** otherwise.
ANS:

```
package checkpassword;
import java.util.*;
public class CheckPassword {
   public static void main(String[] args) {
     Scanner input = new Scanner(System.in);
          System.out.print("Enter a password: ");
          String password = input.nextLine();
          System.out.println(
                    (isValidPassword(password) ? "Valid " : "Invalid ") + "Password");
          }

          public static boolean isValidPassword(String password) {
            final int LENGTH_OF_VALID_PASSWORD = 8;
            final int MINIMUM_NUMBER_OF_DIGITS = 2;
            boolean validPassword =
                    isLengthValid(password, LENGTH_OF_VALID_PASSWORD) &&
                    isOnlyLettersAndDigits(password) &&
                    hasNDigits(password, MINIMUM_NUMBER_OF_DIGITS);
                    return validPassword;
          }
          public static boolean isLengthValid(String password, int validLength) {
                    return password.length() >= validLength;
          }
          public static boolean isOnlyLettersAndDigits(String password) {
                    for (int i = 0; i < password.length(); i++) {
                              if (!Character.isLetterOrDigit(password.charAt(i))) {
                              return false;
                              }
                    }return true;
          }
          public static boolean hasNDigits(String password, int n) {
                    int numberOfDigits = 0;
                    for (int i = 0; i < password.length(); i++) {
                              if (Character.isDigit(password.charAt(i))) {
                              numberOfDigits++;
                              }
                              if (numberOfDigits >= n) {
                              return true;
                    }}
                    return false;
          }}
```

**6.30 (Game: craps) 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. Here are some sample runs.

```
You rolled 5 + 6 = 11
You win
```

```
You rolled 1 + 2 = 3
You lose
```

```
You rolled 4 + 4 = 8
point is 8
You rolled 6 + 2 = 8
You win
```

```
You rolled 3 + 2 = 5
point is 5
You rolled 2 + 5 = 7
You lose
```

ANS:

```
package gamecraps;

public class GameCraps {

   public static void main(String[] args) {
          int point = rollTwoDice();
          int result = getresult(point);
          if (isNaturalOrCraps(result))
          printResult(result);
          else{
        rollTillWinOrLose(result);
          }
          }

          public static int rollDice() {
        return (int)(1 + Math.random() * 6);
          }

          public static int rollTwoDice() {
        int dice1 = rollDice();
        int dice2 = rollDice();
        int sum = dice1 + dice2;
        System.out.println("You rolled " + dice1 + " + " + dice2 + " = " + sum);
        return sum;
          }

          public static int getresult(int point) {
```

```
          switch (point) {
                      case 2  :
                      case 3  :
                      case 12 : point = 0; break;
                      case 7  :
                      case 11 : point = 1; break;
                      }
          return point;
            }

            public static void printResult(int result) {
          if (result == 0)
                      System.out.println("You lose");
          else if (result == 1)
                      System.out.println("You win");
            }

            public static boolean isNaturalOrCraps(int result) {
          return result == 0 || result == 1;
            }

            public static void rollTillWinOrLose(int point) {
          int result;
          do {
                       result = rollTwoDice();
          } while (result != point && result != 7);
                      if (result == 7){
               printResult(0);
             }else{
               printResult(1);  // TODO code application logic here
             }
        }
}
```