Consider the following code fragment. It begins by extracting the substring "3.14" from the string a, and it then converts this substring to the corresponding number, 3.14. Finally, the program reports the result of multiplying this number by 2.0.

    String a = "Pi is 3.14 approx.";   
    String b = a.substring( 6, 10 );   
    double c = Double.parseDouble( b );   
  
    double d = c \* 2.0;   
  
    System.out.println( "The value of d is " + d );

The value of d is 6.28

To convert a double to a String, we can deliberately provoke the automatic conversion that occurs when we concatenate a string and a number by concatenating the floating point number with the empty string. The expression "" + 5.6, for example, evaluates to the string "5.6". Alternatively, we can use the method Double.toString, which is specifically provided for this purpose. This method takes one argument, namely, the floating point value that is to be converted. Then, for example, Double.toString( 5.6 ) evaluates to the string "5.6".

Similar methods are available for integers. In particular, the method Integer.parseInt takes one argument, which must be a string that names an integer, and converts it to an int. So, for example, (Integer.parseInt( "5" ) + 2) evaluates to 7.

Conversely, to convert an int to a string, either concatenate the integer with the empty string or use the one-argument Integer.toString method.

Evaluate each of the following expressions and for each one give its data type.

1. Integer.toString( 5000 )
2. Double.toString( 5.1 ) + 7.5
3. Integer.parseInt( "5" + 5 )
4. Integer.parseInt( "5" ) + 5
5. Double.parseDouble( "5" + "." + "9" + "3" ) + "7"
6. The String "5000".
7. The String "5.17.5".  
   Double.toString( 5.1 ) evaluates to the String "5.1". Then, since one of the operands of the + operator is a string and the other is a number, Java converts the double 7.5 to the String "7.5" and performs a string concatenation.
8. The int 55.  
   Since one of the operands of the + operator is a string and the other is a number, Java converts the int 5 to the String "5" and performs a string concatenation to obtain the String "55". Then Integer.parseInt converts this to the int 55.
9. The int 10.  
   Integer.parseInt converts the String "5" to the int 5. Since both operands of the + operator are numbers, Java performs a numerical addition to obtain the int 10.
10. The String "5.937".  
    By string concatenation, the argument to Double.parseDouble is the String "5.93". Double.parseDouble converts this to the double 5.93. Then, since one of the operands of the + operator is a number and the other is a string, Java converts the double 5.93 back to the String "5.93" and performs a string concatenation.