

Console.java

```

/*package corejava;*/
/**
  An easy interface to read numbers and strings from
  standard input
  @version 1.10 10 Mar 1997
  @author Cay Horstmann
*/

public class Console
{
    /**
     print a prompt on the console but don't print a newline
     @param prompt the prompt string to display
     */
    public static void printPrompt(String prompt)
    { System.out.print(prompt + " ");
      System.out.flush();
    }

    /**
     read a string from the console. The string is
     terminated by a newline
     @return the input string (without the newline)
     */
    public static String readLine()
    { int ch;
      String r = "";
      boolean done = false;
      while (!done)
      { try
        { ch = System.in.read();
          if (ch < 0 || (char)ch == '\n')
          {
              done = true;
          }
          else if ((char)ch != '\r' ) // weird--it used to do \r\n translation
              r = r + (char) ch;
        }
        catch(java.io.IOException e)
        { done = true;
        }
      }
      return r;
    }

    /**
     read a string from the console. The string is
     terminated by a newline
     @param prompt the prompt string to display
     @return the input string (without the newline)
     */
    public static String readLine(String prompt)
    { printPrompt(prompt);
      return readLine();
    }

    /**
     read a char from the console. The char is
     terminated by a newline
     @param prompt the prompt string to display
     @return the input char only (without the newline)
     */
    public static char readChar(String prompt)
    { String s= new String();
      do
      { printPrompt(prompt);
        s=readLine();
        if (s.length()!=1) System.out.println("Only 1 char, please !");
      }while(s.length()!=1);
      return s.charAt(0);
    }
}

```

```
/**
 read an integer from the console. The input is
 terminated by a newline
 @param prompt the prompt string to display
 @return the input value as an int
 @exception NumberFormatException if bad input
 */
public static int readInt(String prompt)
{ while(true)
  { printPrompt(prompt);
    try
    { return Integer.valueOf
      (readLine().trim()).intValue();
    } catch(NumberFormatException e)
    { System.out.println
      ("Not an integer. Please try again!");
    }
  }
}

/**
 read a floating point number with Double precision from console.
 The input is terminated by a newline
 @param prompt the prompt string to display
 @return the input value as a double
 @exception NumberFormatException if bad input
 */
public static double readDouble(String prompt)
{ while(true)
  { printPrompt(prompt);
    try
    { return Double.parseDouble(readLine().trim());
    }
    catch(NumberFormatException e)
    { System.out.println
      ("Not a floating point number. Please try again!");
    }
  }
}

/**
 read a floating point number from the console.
 The input is terminated by a newline
 @param prompt the prompt string to display
 @return the input value as a float
 @exception NumberFormatException if bad input
 */
public static float readFloat(String prompt)
{ while(true)
  { printPrompt(prompt);
    try
    { return Float.parseFloat(readLine().trim());
    }
    catch(NumberFormatException e)
    { System.out.println
      ("Not a floating point number. Please try again!");
    }
  }
}
}
```