

MODULE 7p - Reading Data

CHARACTER INPUT

The following program reads characters from Standard Input and simply copies them to the Standard Output. It makes use of the class `InputStreamReader` whose parent class is `Reader`. There are comments about the merits of using a `Reader` overleaf.

Note that the `read()` method of an `InputStreamReader` object returns an `int` whose value is the coded version of the character read in. A `(char)` cast is required if the associated character is to be used. If there is nothing to read, the value `-1` is returned. Key the program in and try it out.

```
import java.io.InputStreamReader;
import java.io.IOException;

public class ReaderIntro
{ private static final InputStreamReader ISR = new InputStreamReader(System.in);

  public static void main(String[] args)
  { int c;
    try
      { while ((c = ISR.read()) != -1)
          System.out.printf("%c", (char)c);
        }
    catch(IOException e)
      { System.out.printf("IOException Encountered");
        }
    System.out.printf("Data Exhausted");
  }
}
```

EXTRACTING NUMBERS FROM SURROUNDING JUNK

The following program reads characters from Standard Input and extracts sequences of digits delimited by non-digits and converts these sequences into `int` values which are then written out. Key it in and try it out.

```
import java.io.*;

public class ExtractNumbers
{ private static final InputStreamReader ISR = new InputStreamReader(System.in);

  public static void main(String[] args)
  { int i;
    try
      { while ((i = readNum()) != -1)
          System.out.printf("%9d%n", i);
        }
  }
}
```

```
        catch(IOException e)
        { System.out.printf("IOException Encountered");
        }
        System.out.printf("Data Exhausted");
    }

    private static int readNum() throws IOException
    { int c, n=0;
      do
      { c = ISR.read();
        if (c == -1) return -1;
      } while (c<'0' || '9'<c);
      do
      { n = 10*n + c - '0';
        c = ISR.read();
      } while ('0'<=c && c<='9');
      return n;
    }
}
```