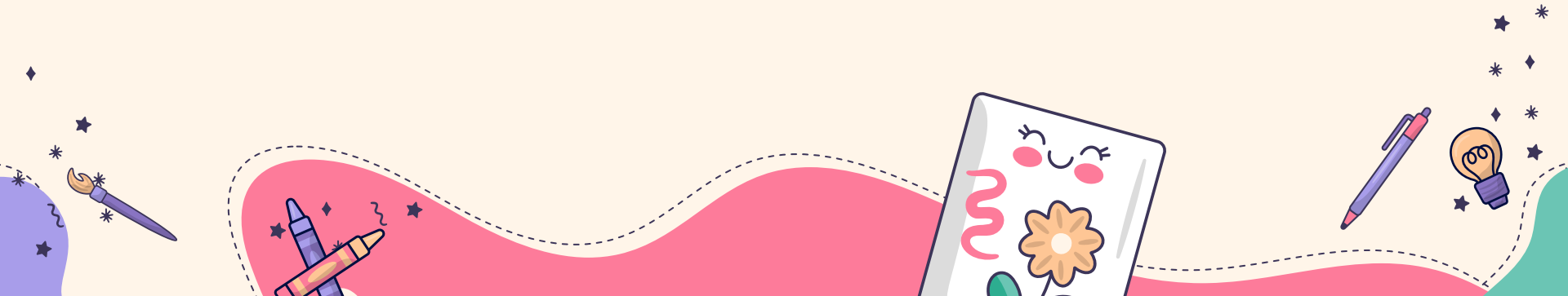
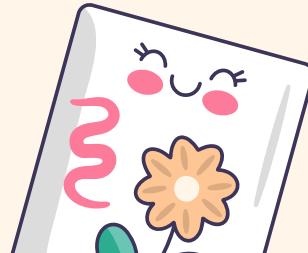


PPBO - 07

# INPUT & OUTPUT



# BASIC I/O



# BASIC I/O

1.

**SCANNER**

2.

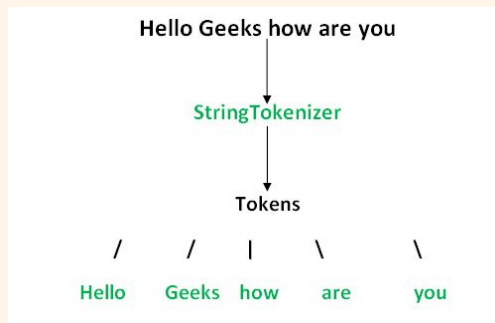
**BufferedReader**

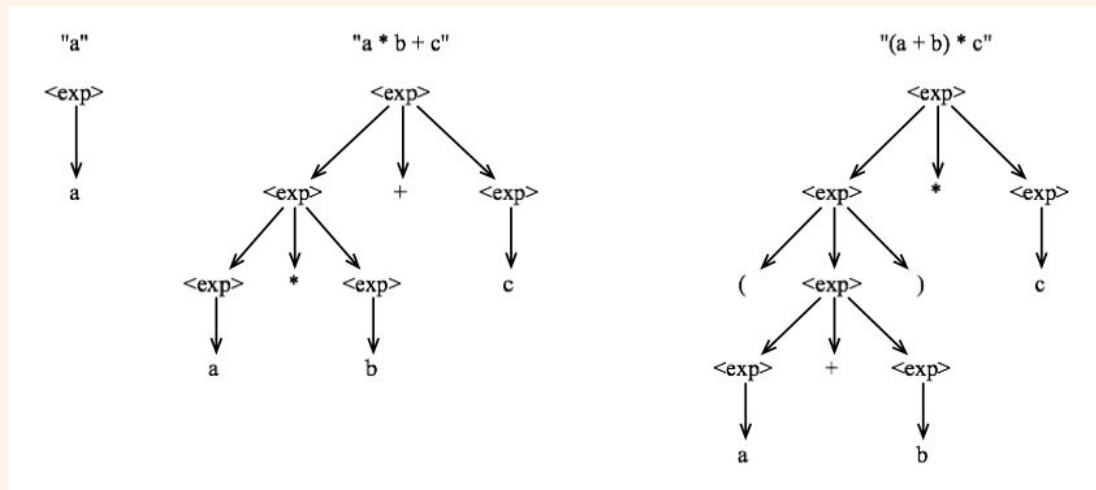
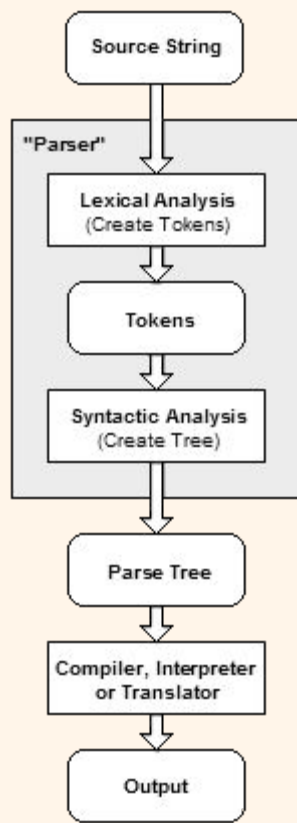
# Scanner

Scanner merupakan salah satu class untuk membaca suatu input yang paling mudah digunakan dan dapat mem-parse tipe data primitif (int, short, byte, float, long, double, dll.) dan string

- Untuk membuat object Scanner, kita dapat memberikan argumen konstruktornya dengan *System.in* untuk membaca dari standard input stream ataupun dengan object File yang ingin dibaca.
- Untuk membaca nilai numerik, gunakan method `nextInt()` untuk integer, `nextFloat()` untuk float, `nextShort()` untuk short, dll.
- Untuk membaca string, gunakan method `nextLine()`
- Untuk membaca 1 character, gunakan method `.next()` yang diikuti dengan `.charAt(0)` → akan mengembalikan character pertama dari input yg diterima.

Object Scanner membaca satu baris penuh input yang diberikan dan membaginya menjadi beberapa token. Token merupakan elemen-elemen kecil yang memiliki makna tersendiri untuk compiler.



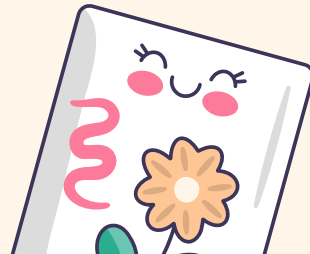


# BufferedReader

BufferedReader merupakan class yang digunakan untuk membaca text dari input stream yang berbasis character. Class ini hanya dapat membaca input sebagai String dengan method `readLine()` tanpa melakukan parse. Oleh karena itu, class ini relatif lebih cepat jika dibandingkan dengan menggunakan Scanner. BufferedReader merupakan turunan dari Class Reader.

Scanner	BufferedReader
<p>Menggunakan regex untuk membaca dan mem-parse input text menjadi tipe data primitif menggunakan berbagai method, seperti <i>nextInt()</i>, <i>nextLong()</i>, <i>nextShort()</i>, dst.</p> <p>Juga dapat membaca String menggunakan method <i>nextLine()</i></p>	<p>Hanya dapat membaca String menggunakan method <i>readLine()</i></p>
<p>Lebih lambat (harus melakukan parse terlebih dahulu)</p>	<p>Lebih cepat (tidak melakukan parse → hanya membaca input stream)</p>
<p>Ukuran buffernya lebih kecil (1 KB)</p>	<p>Ukuran buffernya lebih besar (8 KB)</p>
<p>Lebih cocok untuk membaca input user dari command prompt (karena lebih singkat/sedikit)</p>	<p>Lebih cocok untuk membaca file dengan String yang panjang</p>
<p>Tidak sinkronus (<i>not Synchronized</i>) → Lebih baik digunakan pada kasus yang hanya memerlukan 1 <i>thread</i> saja</p>	<p>Sinkronus (<i>Synchronized</i>) → Dapat digunakan pada kasus yang lebih dari 1 <i>thread</i> (<i>multithread</i>)</p>

# FILE I/O





# FILE I/O

1.

**Low Level File I/O**

2.

**HIGH LEVEL FILE I/O**

# Low Level File I/O

Pada jenis ini, data masih dalam bentuk low-level yaitu berupa stream. Library yang diperlukan:

- File
- FileInputStream
- FileOutputStream

# High Level File I/O

Pada jenis ini, data sudah dalam bentuk high-level yang lebih mudah dimengerti oleh manusia yaitu tipe data primitif.. Library yang diperlukan:

- File
- FileInputStream
- FileOutputStream
- DataInputStream
- DataOutputStream

Data string dapat juga disimpan dalam format ASCII menggunakan library:

- PrintWriter

DEMO





ty



ty



ty



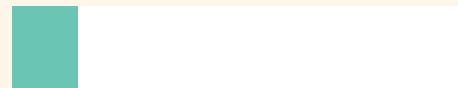
ty



ty



ty



ty