CMPS 12A 5-4-10

Ex. this swap works!

```
static void swap(int[] A, int i, int j){
    int temp;
    temp = A[i];
    A[i] = A[j];
    A[j] = temp;
}
```

In fcn main():

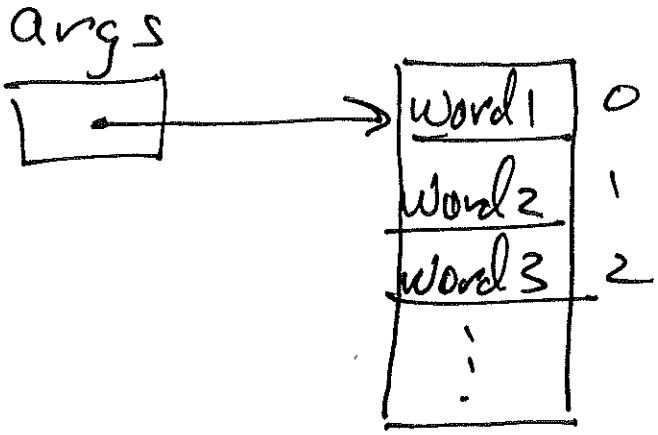
```
int[] list = {9, 8, 7, 6, 5, 4};
swap(list, 1, 5);
swap(list, 2, 3);
PrintArray(list);
//output:          (9 4 6 7 5 8)
swap(list, 1, 6); // throws exception.
```

Command line Arguments

% java myProgram word1 word2 word3 ...

command line arguments

Creates the args array

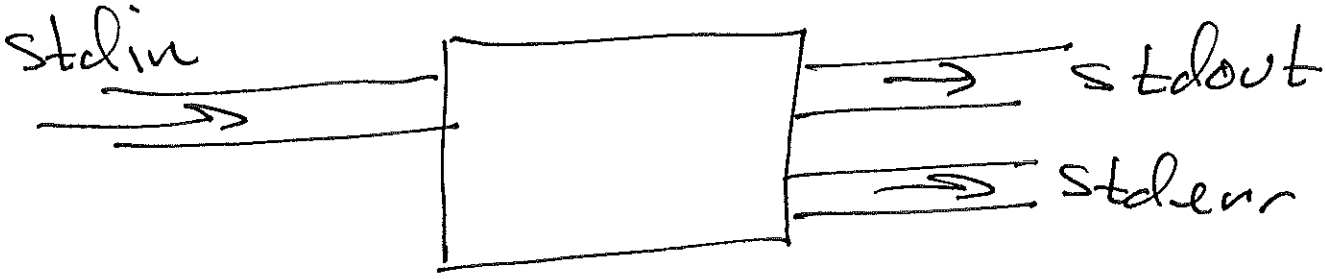


length of args is number of cmd line arguments.

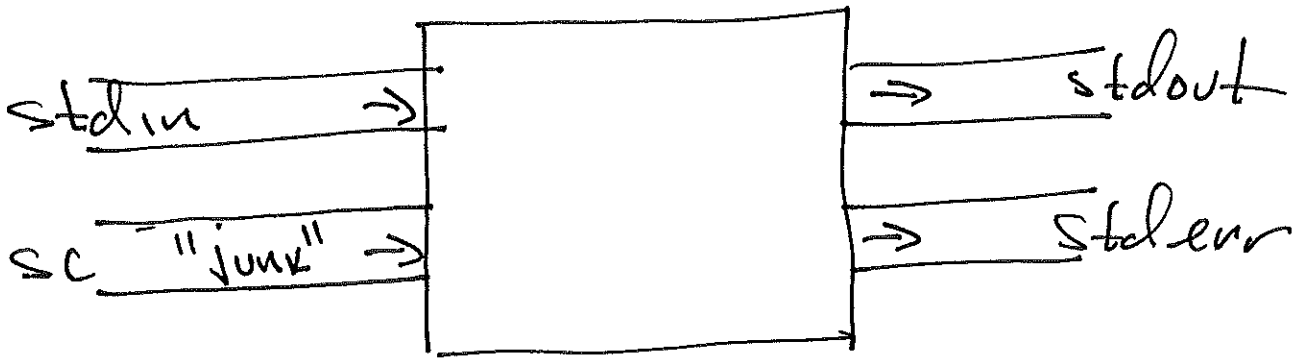
Reading input from a file

see FileIO.java

Recall: 3 data streams



You can open any number of other input streams



Two categories of Java exceptions

- unchecked Exceptions
  - Represent Defects in Program i.e. Programmers fault.
  - subclasses of RuntimeException
  - methods are not required (by compiler) to handle these.
- checked Exceptions
  - Represent error conditions outside Programmers control, not Programmers fault
  - subclasses of Exception (not of RuntimeException
  - methods must handle these exceptions, or won't compile.

How does a method ↓ "handle" an exception?

TWO WAYS:

(1) throw the exception to the calling function by declaring itself to do so

```
... method(...) throws SomeExcep. {
    .
    .
    .
}
```

OR

(2) Catch the exception and handle problem inside the method.

this (option 2) is implemented by try – catch :

```
try {
    some of. that might throw
    an exception
} catch( ExceptionType ExceptionVar ){
    code to recover from error
    condition, or quit in some
    special way
}
```

or

```
try {
    code that might throw several
    different exceptions
} catch (ExceptionType1 e1) {
    code that deals with e1
} catch (ExceptionType2 e2) {
    code that deals with e2
} finally {
    code that handles anything
    else that may have happened
}
```

note:

- try must have at least 1 catch
- braces {..} are required.

Ex. TryCatch.java