Examples

	Valid	Invalid
happy	✓	
happy	✓	
$happy$	✓	
2happy		✓
happy2	✓	
HAPPY	✓	
A_Long_Variable_Name	✓	
class		✓
2class		✓
class2	✓	
A Long Variable Name		✓
"A Long Variable Name"		✓

Most programs use a variety of different data to accomplish their task. Data is categorized and organized by types. A data type is defined by the way the data will be stored in memory, i.e. amount of memory used and the way data is encoded. Most of these details, i.e. exactly how a number is stored in memory for instance, are outside the scope of this course.

However we do need to know what the basic data types are:

DATA TYPE	USED TO STORE
int	INTEGERS
double	REAL NUMBERS (UP TO SOME # DIGITS)
String	TEXT
char	SINGLE CHARACTERS
boolean	true, false

THERE ARE MANY OTHER BUILT IN DATA TYPES.

ACTUALLY ALL JAVA DATA TYPES FALL INTO TWO CATEGORIES

PRIMITIVE TYPES:
- byte, short, int, long, float, double (NUMERIC)
- char
- boolean

CLASS TYPES
- String
- MANY OTHERS, INCLUDING THOSE DEFINED BY PROGRAMMERS.

LITERALS, ALSO CALLED CONSTANTS, ARE THE PROGRAM REPRESENTATIONS OF THE VALUES OF A DATA TYPE.

EXAMPLES:

int	6, -57, 3954
double	1.23, 3.45, -0.0001
String	"Happy", "Hello, World!"

boolean	true, false
char	'a', 'b', 'A', ';'

IN ADDITION TO KEYWORDS, IDENTIFIERS, AND LITERALS, A JAVA PROGRAM CONTAINS OPERATORS AND PUNCTUATION SYMBOLS.

AN OPERATOR IS SOMETHING LIKE + USED TO ADD NUMBERS. THUS

$$6 + 5$$

FORMS AN EXPRESSION WHICH CAN BE EVALUATED. TO THE int VALUE 11.

MANY OPERATORS HAVE AN EXTENDED MEANING THAT CAN BE SPECIFIED BY ITS CONTEXT. FOR INSTANCE + CAN BE USED IN:

"Hello" + ", World!"

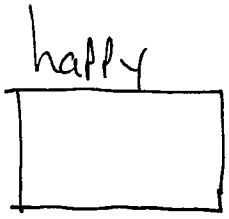
TO FORM A String EXPRESSION THAT EVALUATES TO THE String VALUE "Hello, World!"

PUNCTUATION SYMBOLS ARE THINGS LIKE

{, }, (,), [,], ;, ,, ", ETC.

THEIR MEANINGS WILL BECOME CLEAR AS THE COURSE PROGRESSES.

A VARIABLE IS AN IDENTIFIER USED TO REFER TO A DATA VALUE STORED IN COMPUTER MEMORY



- variable name
- memory location.

VARIABLES ARE ESTABLISHED BY DECLARATION STATEMENTS.

EXAMPLES

```
int i;
int j, k;
double average
String word1, word2;
```

NOTE THAT DECLARATION STATEMENTS, LIKE ALL STATEMENTS IN JAVA, MUST END IN A SEMICOLON ;

VARIABLES CAN BE INITIALIZED AT THE SAME TIME THEY ARE DECLARED.

Ex.

```
int i = 6, j = 7, k;
double ammount = 0.0;
String word = "Hello";
boolean happy = true, sad = false;
```

VARIABLES CAN ALSO BE DECLARED AND INITIALIZED SEPARATELY.

Ex.

```
// HelloWorld3.java
class HelloWorld3 {
    Public static void main(String[] args){
        String word1;
        String word2;
        String sentence;
        word1 = "Hello";
        word2 = ", World!";
        sentence = word1.concat(word2);
        System.out.Println(sentence);
    }
}
```

MEMORY:

word1	word2	sentence
Hello	, World!	Hello, World!

IN THIS CONTEXT, = IS THE ASSIGNMENT OPERATOR. WE USE IT TO STORE A VALUE IN A VARIABLE.

THE METHOD concat( ) BELONGS TO THE String class, AND IS USED TO CONCATENATE STRINGS. AN ALTERNATE WAY TO CONCATENATE STRING VALUES IS WITH THE + OPERATOR AS IN:

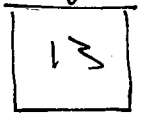
```
sentence = word1 + word2;
```

HOW DOES THE COMPILER TELL THE DIFFERENCE BETWEEN THE DIFFERENT MEANINGS OF THE + SYMBOL?

EX

```
int i
i = 6 + 7;
```

MEMORY:



IN THE FIRST EXAMPLE, THE OPERANDS TO + ARE OF TYPE String, AND IN THE SECOND, THE OPERANDS ARE OF TYPE int. WE SAY THE SYMBOL + IS OVERLOADED.

MOST PROGRAMS INPUT DATA FROM A USER AS WELL AS Print OUTPUT.

We can use the Scanner class found in the library java.util to take input from the keyboard.

Ex.

```
// HelloWorld4.java

import java.util.*;

class HelloWorld4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence1 = "Hello, World!";
        String sentence2 = "My name is ";
        String name;

        System.out.print("Please type your first name: ");
        name = sc.next();
        System.out.println(sentence1);
        System.out.print(sentence2);
        System.out.println(name);
    }
}
```

OUTPUT:

```
% java HelloWorld4
Please type your first name: Pat
Hello, World!
My name is Pat
%
```

java.util is a standard java library package containing utilities for many common tasks. You include it in your program by placing

```
import java.util.*;
```

before the class definition. One can also import a single class, as opposed to the whole library by doing:

```
import java.util.Scanner;
```

Thus the symbol * in this context just means everything.

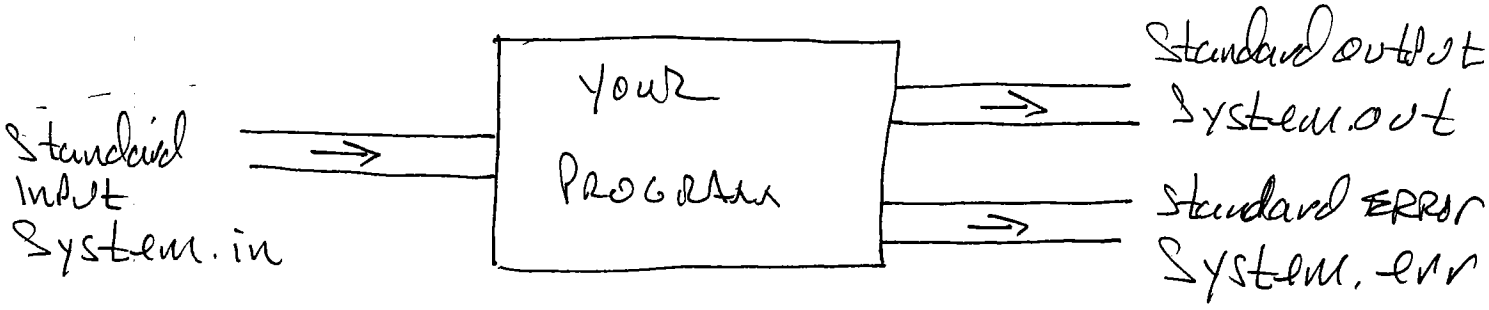
The line

```
Scanner sc = new Scanner(System.in);
```

declares sc to be a variable of type Scanner, and initializes it to refer to terminal input, i.e. a proxy for the keyboard.

Every invocation of a Java program comes equipped with three standard data streams, which are literally sequences of characters

Stream	Java Reference	Default
• Standard input:	System.in	keyboard
• Standard output:	System.out	screen
• Standard error:	System.err	screen



THESE STREAMS REFER TO KEYBOARD AND SCREEN (RESP.) BY DEFAULT BUT CAN BE MADE TO REFER TO OTHER DEVICES.

OTHER DATA STREAMS CAN BE CREATED BY THE PROGRAMMER IN VARIOUS WAYS (MORE LATER.)

THE LINE

```
name = sc.next();
```

READS THE NEXT WHOLE STRING (i.e. WITHOUT SPACES) FROM STANDARD INPUT AND ASSIGNS IT TO THE STRING VARIABLE name.

MEMORY