

Basic Computer Science and Java Concepts that everyone should understand

1. Variables

- a. Variables in Java represent a storage location in the computers memory
 - i. Variables can change over time
 - ii. Variables have a specific type associated with them (see next section)

2. Java Types

- a. Two flavors:
 - i. Primitive
 - 1. int, float, double, boolean...
 - ii. Reference (objects or composite)
 - 1. String, ...

Primitive Types contain just the actual data, nothing more, e.g.:

```
int age = 5; // the variable age is of type int and contains just the number 5, nothing else
```

Reference Types contain the actual data and other information, e.g.:

```
String name = "Amanda"; // the variable name contains the data "Amanda" and other data as well
```

Primitive types in Java and their ranges:

Type	Contains	Default	Size	Range
boolean	true or false	false	1 bit	NA
char	Unicode character	\u0000	16 bits	\u0000 to \uFFFF
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
long	Signed integer	0	64 bits	-9223372036854775808 to 9223372036854775807
float	IEEE 754 floating point	0.0	32 bits	$\pm 1.4E-45$ to $\pm 3.4028235E+38$
double	IEEE 754 floating point	0.0	64 bits	$\pm 4.9E-324$ to $\pm 1.7976931348623157E+308$

What about String?

String is **not** a primitive type; it is an object (Reference type)

More information on primitive and reference types in the text and available at:

<http://download.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

Notes:

Java is strongly typed, why?

To avoid Runtime errors that can cause erroneous behavior, e.g.

```
...
int age = 5; // variables of type int can only be whole numbers

double distance = 23.259; // the distance between Los Angeles and Seal Beach

age = distance; // this doesn't make sense for several reasons, hence a compiler error occurs, e.g.:
```

```
public class misMatch
{
    public static void main(String[] args)
    {
        int age = 5;
        double distance = 23.259;

        age = distance; // ???!?????
    }
}
```

```
$ javac misMatch.java
misMatch.java:8: possible loss of precision
found   : double
required: int
age = distance; // ???!?????
          ^
1 error
```

Note that the above error, "possible loss of precision" is a result of trying to store a value of type double into a variable of type int

Why does the following cause an error during compilation?

```
float pi = 3.14;
```

3. Assignment statements

Assignment statements in Java store values in variables and the contents of one variable into another, e.g.:

```
public class assignStatement
{
    public static void main(String[] args)
    {
        // Calculate the area of a circle

        double pi = 3.1415926535;

        int radius = 3;
        double area = pi * radius * radius;

        System.out.println("The area of a circle, with radius 3 is: " + area);
    }
}
```

Sample run:

```
$ java assignStatement
The area of a circle, with radius 3 is: 28.2743338815
```

Notes:

- Variables can be of different types, e.g. pi is of type double and radius is of type int
- Assignment statements take a value or result from the right side of the equals sign and stores the result or value into a variable on the left side:

```
int radius = 3;
```

Stores the integer value 3 into the variable radius, pictorially:

```
int radius = 3;
  ↖         ↗
```

- The following line takes the product of three variables and stores the result into the variable *area*

```
double area = pi * radius * radius;
```

Note that mixing (multiplying) variables of type int and type double and storing the result into a variable of type double is legal, since there isn't loss of precision resulting from type mismatch.