

# JAVA fundamentals – class notes (dated 2014)

# Sample code

```
/*  
  This is a simple Java program.  
  Call this file "Example.java".  
*/  
  
class Example {  
  // Your program begins with a call to main().  
  public static void main(String args[]) {  
    System.out.println("This is a simple Java program.");  
  }  
}
```

```
/*  
    Here is another short example.  
    Call this file "Example2.java".  
*/  
class Example2 {  
    public static void main(String args[]) {  
        int num; // this declares a variable called num  
        num = 100; // this assigns num the value 100  
        System.out.println("This is num: " + num);  
  
        num = num * 2;  
        System.out.println("The value of num * 2 is ");  
        System.out.println(num);  
    }  
}
```

```
/*  
Demonstrate the if.  
  
Call this file "IfSample.java".  
*/  
class IfSample {  
    public static void main(String args[]) {  
        int x, y;  
  

```

```
/*
```

Demonstrate the for loop.

Call this file "ForTest.java".

```
*/
```

```
class ForTest {  
    public static void main(String args[]) {  
        int x;  
  
        for(x = 0; x<10; x = x+1)  
            System.out.println("This is x: " + x);  
    }  
}
```

```
/*  
  Demonstrate a block of code.  
  
  Call this file "BlockTest.java"  
*/  
class BlockTest {  
  public static void main(String args[]) {  
    int x, y;  
    y = 20;  
  
    // the target of this loop is a block  
    for(x = 0; x<10; x++) {  
      System.out.println("This is x: " + x);  
      System.out.println("This is y: " + y);  
      y = y - 2;  
    }  
  }  
}
```

```
// Compute distance light travels using long variables.
class Light {
    public static void main(String args[]) {
        int lightspeed;
        long days;
        long seconds;
        long distance;

        // approximate speed of light in miles per second
        lightspeed = 186000;

        days = 1000; // specify number of days here

        seconds = days * 24 * 60 * 60; // convert to seconds

        distance = lightspeed * seconds; // compute distance

        System.out.print("In " + days);
        System.out.print(" days light will travel about ");
        System.out.println(distance + " miles.");
    }
}
```

# Data type - Integer

Java defines four integer types: **byte**, **short**, **int**, and **long**. All of these are signed, **positive** and negative values. Java does not support unsigned, positive-only integers. Many other computer languages, including C/C++, support both signed and unsigned integers.

The width and ranges of these integer types vary widely, as shown in this table:

Name	Width	Range
<b>long</b>	64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
<b>int</b>	32	-2,147,483,648 to 2,147,483,647
<b>short</b>	16	-32,768 to 32,767
<b>byte</b>	8	-128 to 127



# Data type – Float

## Floating-Point Types

Floating-point numbers, also known as *real numbers*, are used when evaluating expressions that require fractional precision.

Name	Width in Bits	Approximate Range
<b>double</b>	<b>64</b>	<b>4.9e<sup>-324</sup> to 1.8e<sup>+308</sup></b>
<b>float</b>	<b>32</b>	<b>1.4e<sup>-045</sup> to 3.4e<sup>+038</sup></b>

# Data type - Characters

In Java, the data type used to store characters is **char**. However, C/C++ programmers beware: **char in Java is not the same as char in C or C++**. In C/C++, **char is an integer** type that is 8 bits wide. This is *not the case in Java*. Instead, Java uses Unicode to represent characters. Unicode defines a fully international character set that can represent all of the characters found in all human languages.

// Demonstrate char data type.

```
class CharDemo {
public static void main(String args[]) {
char ch1, ch2;
ch1 = 88; // code for X
ch2 = 'Y';
System.out.print("ch1 and ch2: ");
System.out.println(ch1 + " " + ch2);
}
}
```

This program displays the following output:

ch1 and ch2: XY

Notice that **ch1 is assigned the value 88, which is the ASCII (and Unicode) value that** corresponds to the letter *X*. *As mentioned, the ASCII character set occupies the first 127 values in the Unicode character set.*

```
// char variables behave like integers.  
class CharDemo2 {  
public static void main(String args[]) {  
char ch1;  
ch1 = 'X';  
System.out.println("ch1 contains " + ch1);  
ch1++; // increment ch1  
System.out.println("ch1 is now " + ch1);  
}  
}
```

The output generated by this program is shown here:

ch1 contains X

ch1 is now Y

In the program, **ch1 is first given the value X. Next, ch1 is incremented. This results in ch1 containing Y, the next character in the ASCII (and Unicode) sequence.**

# Boolean

Java has a simple type, called **boolean**, for logical values. It can have only one of two possible values, **true** or **false**. This is the type returned by all relational operators, such as **a < b**. **boolean** is also the type *required by the conditional expressions that govern the* control statements such as **if** and **for**.

```
// Demonstrate boolean values.
class BoolTest {
public static void main(String args[]) {
boolean b;
b = false;
System.out.println("b is " + b);
b = true;
System.out.println("b is " + b);
// a boolean value can control the if statement
if(b) System.out.println("This is executed.");
b = false;
if(b) System.out.println("This is not executed.");
// outcome of a relational operator is a boolean value
System.out.println("10 > 9 is " + (10 > 9));
}}
```

# Class – Its general form

A class is declared by use of the **class keyword**.

**The classes that have been used up**

to this point are actually very limited examples of its complete form. Classes can (and usually do) get much more complex. The general form of a **class definition is shown here:**

```
class classname {  
    type instance-variable1;  
    type instance-variable2;  
    // ...  
    type instance-variableN;  
    type methodname1(parameter-list) {  
        // body of method  
    }  
    type methodname2(parameter-list) {  
        // body of method  
    }
```

```
}  
// ...  
type methodName(parameter-list) {  
// body of method  
}  
}
```

*/\*The data, or variables, defined within a **class** are called **instance variables**. The code is contained within *methods*.\*/*

# Thanks