Primitive Data Types

Data Type	Memory	Range
INTEGER		
long	**64-bits**	-2^{64} to $+2^{64}$ -1
int	**32-bits**	-2^{32} to $+2^{32}$ -1
short	**16-bits**	-2^{16} to $+2^{16}$ -1
byte	**8-bits**	-2^{8} to $+2^{8}$ -1
FLOATING POINT		
double	**64-bits**	1.7e-308 to 1.7e +308
float	**32-bits**	3.4e-038 to 3.4 e+038
CHARACTERS		
char	**16-bits**	0 to 65535
BOOLEANS		
boolean		true or false

Data Types (default values)

Data Type	**Default Value (for fields)**
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0
char	'\u0000'
boolean	false

Java Operators

Arithmetic	Increment/ Decrement	Assignment	Logical	Comparison	Bitwise
+	++	=		`x < y`	`~x`
-	--	+=, -=, /=,%=	&&	`x <= y`	`x & y`
*		`&=, |=, ^=`	||	`x > y`	`x | y`
/		`<<=, >>=, >>>=`	!	`x > y`	`x ^ y`
%				`x >= y`	`x << y`
-(unary minus)				`x == y`	`x >> y`
+(unary plus)				`x != y`	`x >>> y`

Control Statements

Conditional if-else Statement:

```
if (condition) {
   // statement sequence
} else {
 // other statement sequence
}
```

Repetitive Loop Statements:

***for* loop**

```
for (i = 0; i < 10; i++) {
      //statement sequence
      }
```

Enhanced *for* loop (added with Java 5.0)

```
public class bitx
 {
public static void main(String s[])
{
 int x[]={10,23,45,67,88};
for(int a:x)
{
        System.out.println(a);
}
}
}
```

***while* loops**

```
while (i < 5) {
   //statement sequence
}
```

***do/while* loops**

```
do{
      //statement sequence
   } while (condition);
```

- ***switch* statement**

```
switch (i) {
    case 1:
// statement sequences
   break;
     case 2:
// statement sequences
   break;
     default:
 // default statement sequences

}
```