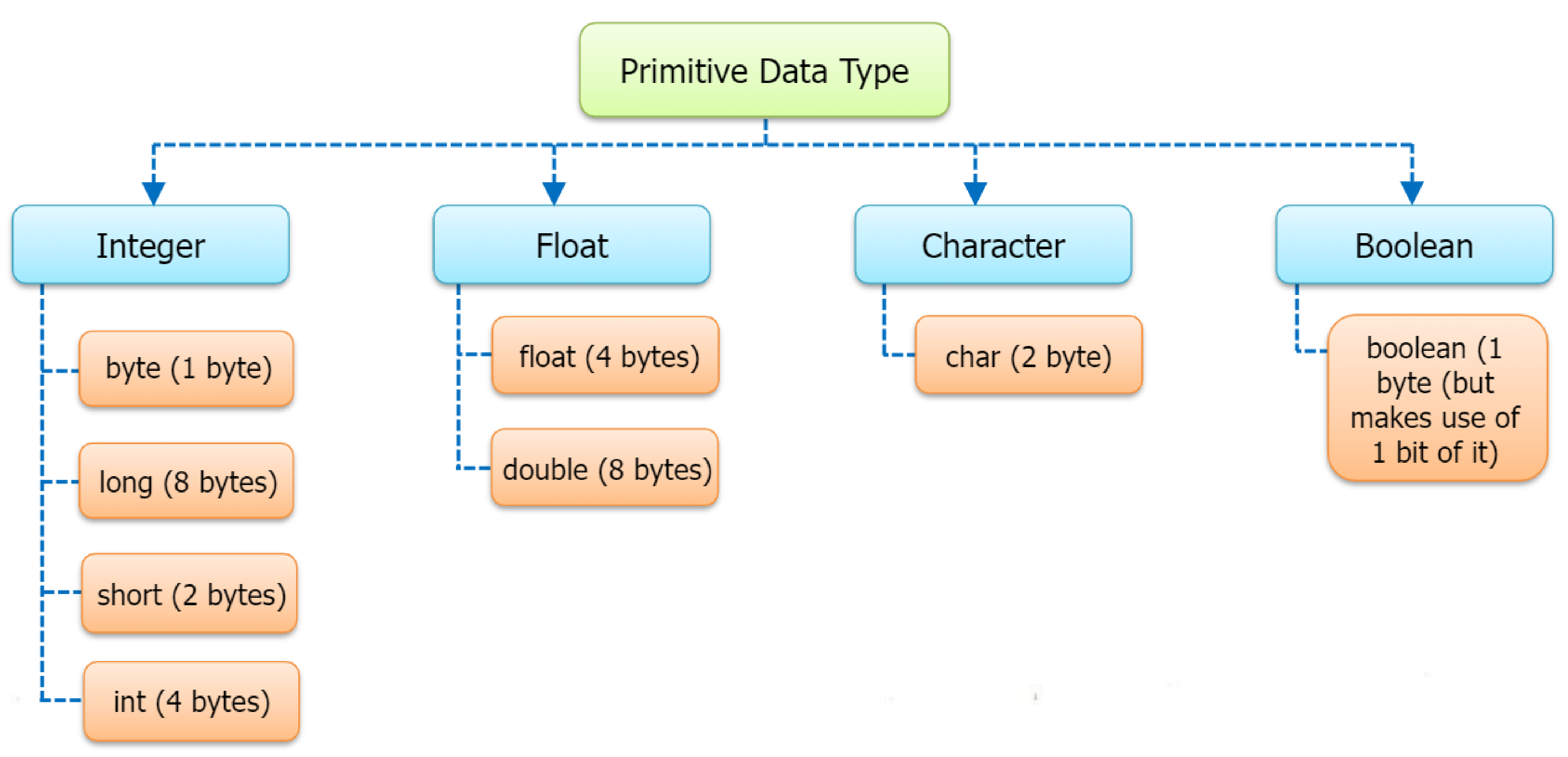
```
hp body_class
'fb-root"></div>
(function(d, s, id) {
, fjs = d.getElementsByTagName(s)[0];
getElementById(id)) return;
.createElement(s); js.id = id;
c = "//connect.facebook.net/en_US/sdk.js#xfb
arentNode.insertBefore(js, fjs);
ent, 'script', 'facebook-jssdk'));</script>
="page" class="site">
 class="skip-link screen-reader-text" href="

eader id="masthead" class="site-header" rol
  <div class="site-branding">
      <div class="navBtn pull-left">
          <?php if(is_home() && $xpanel['ho
          <a href="#" id="openMenu"><i clas
          <?php } else { ?>
          <a href="#" id="openMenu2"><i cl
          <?php } ?>
      </div>
      <div class="logo pull-left">
          <a href="<?php echo esc_url( ho
              <img src="<?php echo $xpane
          </a>
      </div>
      <div class="search-box hidden-xs h
          <?php get_search_form(); ?>
      </div>
      <div class="submit-btn hidden-xs h
          <a href="<?php echo get_page_
      </div>
      <div class="user-info pull-right
          <?php
          if ( is_user_logged_in() ) {
                  current_user;
```

TODAY'S AGENDA

- Data Types
- String Concatenation
- Operators in Java
- If Else

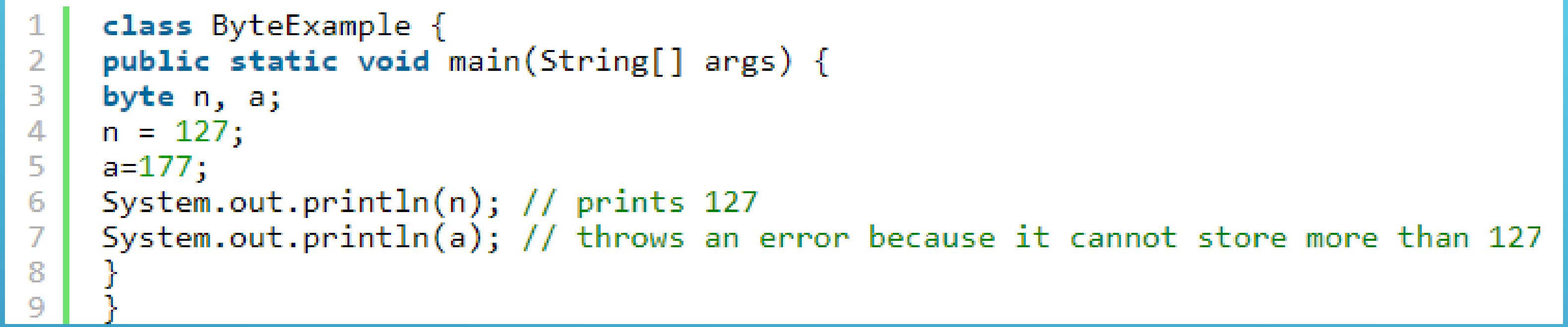
Data Types



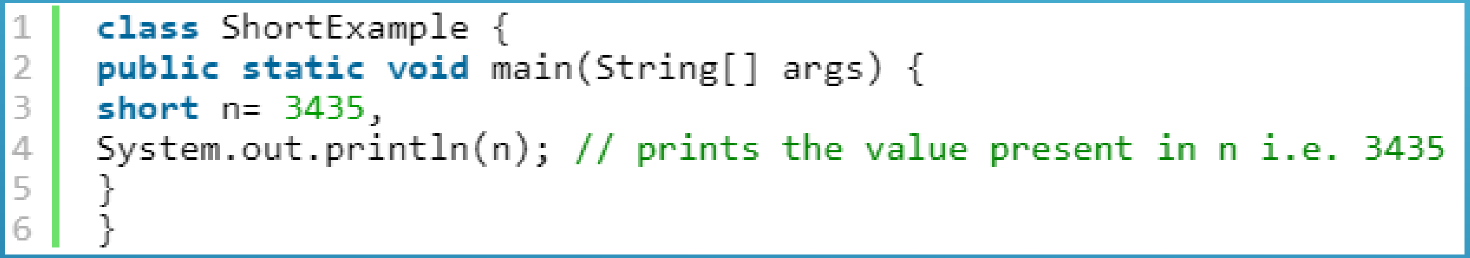
Primitive Data Type

- Integer
  - byte (1 byte)
  - long (8 bytes)
  - short (2 bytes)
  - int (4 bytes)
- Float
  - float (4 bytes)
  - double (8 bytes)
- Character
  - char (2 byte)
- Boolean
  - boolean (1 byte (but makes use of 1 bit of it)

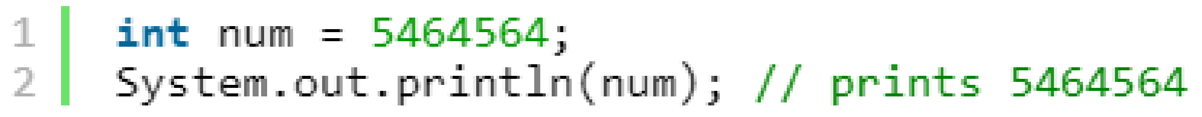
Data Types



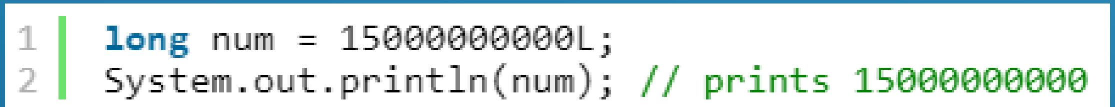
```java
class ByteExample {
public static void main(String[] args) {
byte n, a;
n = 127;
a=177;
System.out.println(n); // prints 127
System.out.println(a); // throws an error because it cannot store more than 127
}
}
```



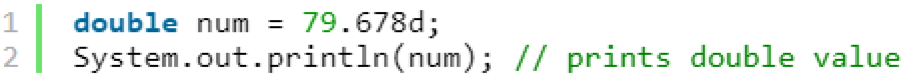
```java
class ShortExample {
public static void main(String[] args) {
short n= 3435,
System.out.println(n); // prints the value present in n i.e. 3435
}
}
```



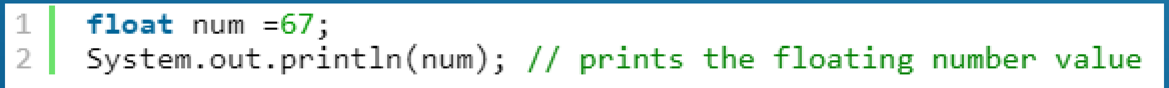
```java
 int num = 5464564;
 System.out.println(num); // prints 5464564
```



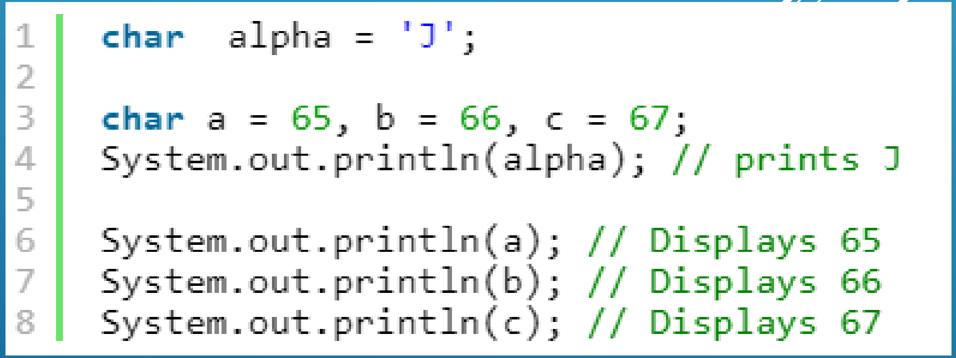
```java
long num = 15000000000L;
System.out.println(num); // prints 15000000000
```



```java
double num = 79.678d;
System.out.println(num); // prints double value
```



```java
float num =67;
System.out.println(num); // prints the floating number value
```

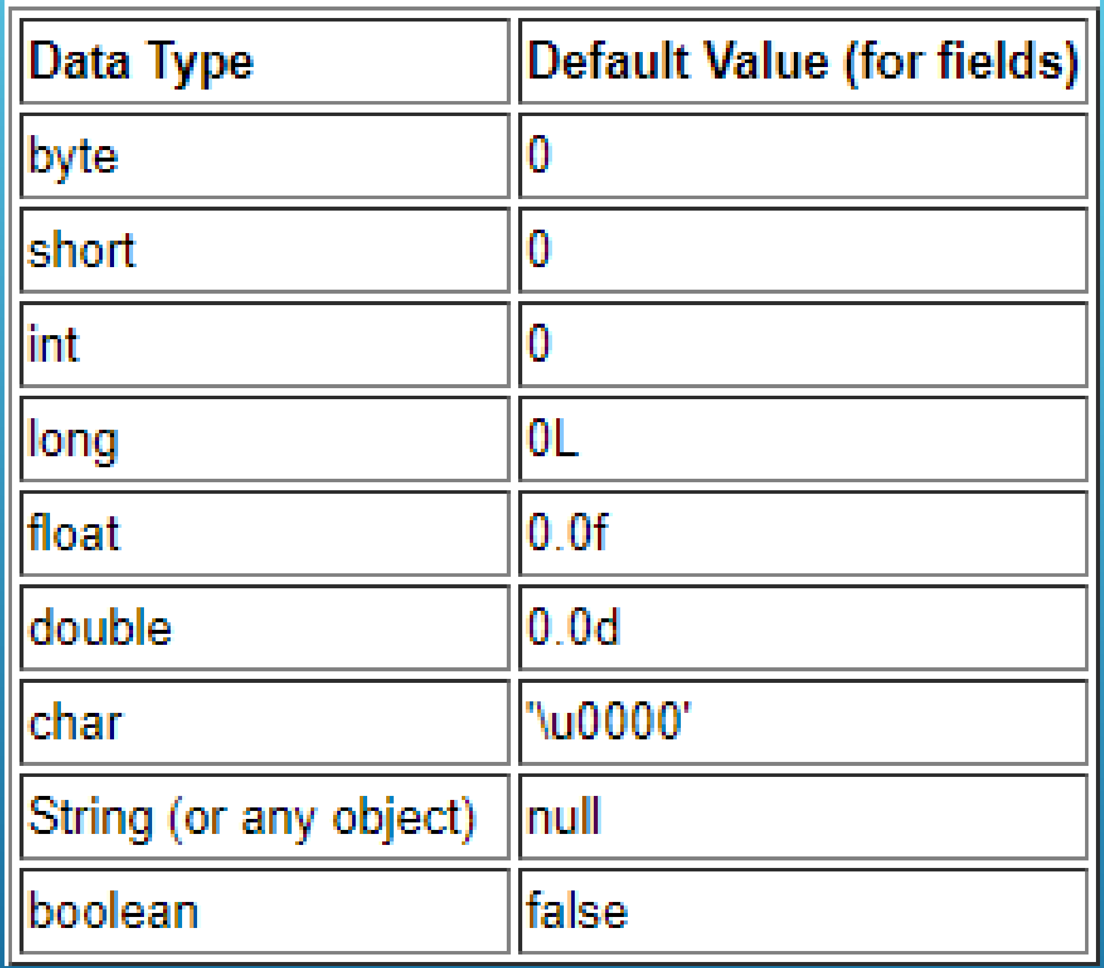


```java
char  alpha = 'J';

char a = 65, b = 66, c = 67;
System.out.println(alpha); // prints J

System.out.println(a); // Displays 65
System.out.println(b); // Displays 66
System.out.println(c); // Displays 67
```

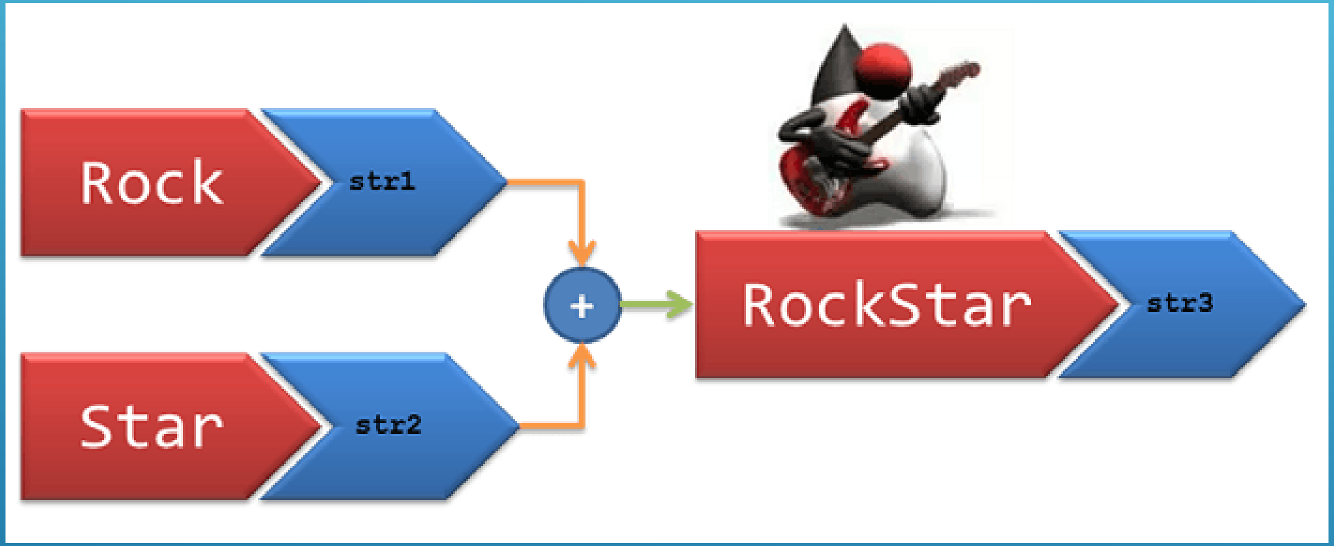
Default Values



Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any object)	null
boolean	false

String Concatenation

Concatenation is the process of appending one string to the end of another string. You concatenate strings by using the + operator.



WHAT ARE OPERATORS IN JAVA AND ITS TYPES?

- Operators are the constructs which can manipulate the values of the operands. Consider the expression 2 + 3 = 5, here 2 and 3 are operands and + is called operator. In this article on Java operators, the goal is to get you the expertise required to get started and work with operators in Java.
- Java supports the following types of operators:
    - Arithmetic Operators
    - Assignment Operators
    - Logical Operators
    - Relational Operators
    - Unary Operators

ARITHMETIC OPERATORS IN JAVA

- Arithmetic Operators are used to perform mathematical operations like addition, subtraction, etc. Assume that A = 10 and B = 20 for the below table.

Operator	Description	Example
+ Addition	Adds values on either side of the operator	A+B=30
– Subtraction	Subtracts the right-hand operator with left-hand operator	A-B=-10
* Multiplication	Multiplies values on either side of the operator	A*B=200
/ Division	Divides left hand operand with right hand operator	A/B=0
% Modulus	Divides left hand operand by right hand operand and returns remainder	A%B=0

Example:

```
package Arithmetic;


public class ArithmeticOperators {
    public static void main(String[] args) {
        int A = 10;
        int B = 20;
        System.out.println(A + B);
        System.out.println(A - B);
        System.out.println(A * B);
        System.out.println(A / B);
        System.out.println(A % B);
    }
}
```

Output:

```
30
-10
200
0
10
```

ASSIGNMENT OPERATORS IN JAVA

- An Assignment Operator is an operator used to assign a new value to a variable. Assume A = 10 and B = 20 for the below table.

Operator	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b
+=	It adds right operand to the left operand and assigns the result to left operand	c += a
-=	It subtracts right operand from the left operand and assigns the result to left operand	c -= a
*=	It multiplies right operand with the left operand and assigns the result to left operand	c *= a
/=	It divides left operand with the right operand and assigns the result to left operand	c /= a
%=	It takes modulus using two operands and assigns the result to left operand	c %= a
^=	Performs exponential (power) calculation on operators and assign value to the left operand	c ^= a

Example:

```
package Operators ;

public class JavaOperators {
    public static void main(String[] args) {
        int a = 10;
        int b=20;
        int c;
        System.out.println(c = a); // Output =10
        System.out.println(b += a);// Output=30
        System.out.println(b -= a);// Output=20
        System.out.println(b *= a);// Output=200
        System.out.println(b /= a);// Output=2
        System.out.println(b %= a);// Output=0
        System.out.println(b ^= a);// Output=0
       }
}
```

RELATIONAL OPERATORS IN JAVA

- These operators compare the values on either side of them and decide the relation among them. Assume A = 10 and B = 20.

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(A == B) is not true
!=	If the values of two operands are not equal, then condition becomes true.	(A != B) is true
>	If the value of the left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true
<	If the value of the left operand is less than the value of right operand, then condition becomes true.	(a < b) is true
>=	If the value of the left operand is greater than or equal to the value of the right operand, then condition becomes true.	(a >= b) is not true
<=	If the value of the left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true

Example:

```
package Operators;

public class JavaOperators {
    public static void main(String[] args) {
        int a = 10;
        int b=20;
        System.out.println(a == b); // returns false
because 10 is not equal to 20
        System.out.println(a != b); // returns true
because 10 is not equal to 20
        System.out.println(a > b); // returns false
        System.out.println(a < b); // returns true
             System.out.println(a >= b); // returns false
        System.out.println(a <= b); // returns true
       }

}
```

LOGICAL OPERATORS IN JAVA

- Logical operators are used to check whether an expression is true or false

Operator	Description	Example
&& (and)	True if both the operands is true	a<10 && a<20
| | (or)	True if either of the operands is true	a<10 | | a<20
! (not)	True if an operand is false (complements the operand)	!(x<10 && a<20)

Example:

```
package Operators;

public class JavaOperators {
      public static void main(String[] args) {
            int a = 10;
        System.out.println(a<10 & a<20);  //returns false
        System.out.println(a<10 || a<20); //returns true
        System.out.println(!(a<10 &  a<20)); //returns true
          }
}
```

UNARY OPERATOR IN JAVA

- Unary operators are the one that needs a single operand and are used to increment a value, decrement or negate a value.

Operator	Description	Example
++	increments the value by 1. There is post-increment and pre-increment operators	a++ and ++a
—	decrements the value by 1. There is post decrement and pre decrement operators	a– or –a
!	invert a boolean value	!a

Example:

```
package Operators;

public class JavaOperators {
      public static void main(String[] args) {
            int a = 10;
            boolean b=true;
        System.out.println(a++);  //returns 11
        System.out.println(++a);
        System.out.println(a--);
        System.out.println(--a);
        System.out.println(!b); // returns false
          }
}
```

IF STATEMENTS

Java has the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false

THE IF STATEMENT

- Use the if statement to specify a block of Java code to be executed if a condition is true.

Syntax

```
if (condition) {
// block of code to be executed if the condition
is true
}
```

Example

```
if (20 > 18) {
System.out.println("20 is greater than 18");
}

int x = 20;
int y = 18;
if (x > y) {
System.out.println("x is greater than
y");
}
```

THE ELSE STATEMENT

- Use the else statement to specify a block of code to be executed if the condition is false.

Syntax

```
if (condition) {
// block of code to be executed if the condition is true
}else {
// block of code to be executed if the condition is false
}
```

Example

```
int time = 20;
if (time < 18) {
System.out.println("Good day.");
} else {
System.out.println("Good evening.");
}
// Outputs "Good evening."
```

THE ELSE IF STATEMENT

- Use the else if statement to specify a new condition if the first condition is false.

Syntax

```
if (condition1) {
// block of code to be executed if condition1 is true
} else if (condition2) {
// block of code to be executed if the condition1 is false and condition2 is true
} else {
// block of code to be executed if the condition1 is false and condition2 is false
}
```

Example

```
int time = 22;
if (time < 10) {
System.out.println("Good morning.");
} else if (time < 20) {
System.out.println("Good day.");
} else {
System.out.println("Good evening.");
}
 // Outputs "Good evening."
```

HOMEWORK

if Else:

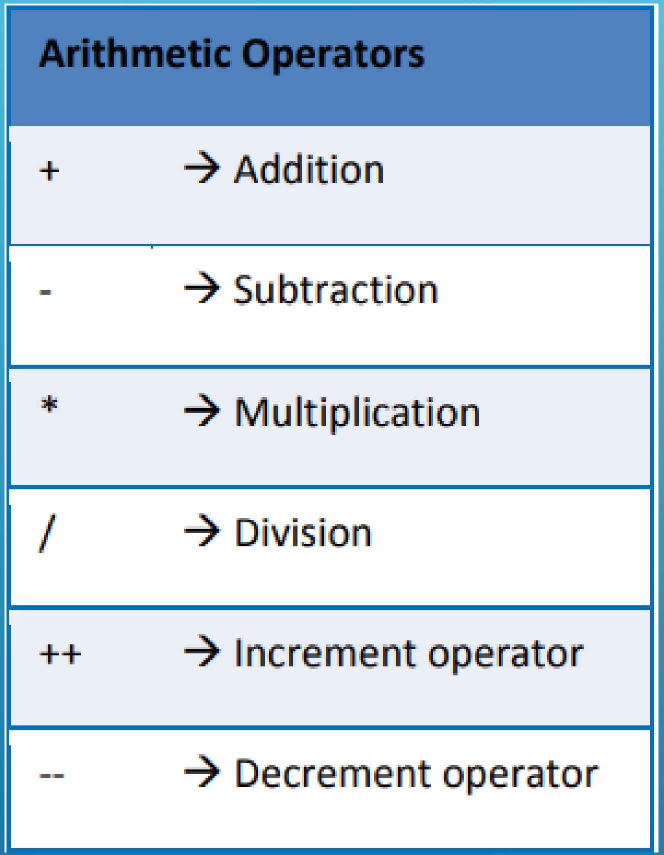
- 1. Write a program to check if a candidate is eligible for voting or not. (Hint: Check age)

- 2. Write a program to check if the number is positive or negative.

- 3. Extend the previous program to check whether the given number is positive, zero or negative. (Hint: use if-else conditions)

- 4. Write a program to check given number is even or odd.
  (Hint: use % operator)

HOMEWORK



Arithmetic Operators	
+	→ Addition
-	→ Subtraction
*	→ Multiplication
/	→ Division
++	→ Increment operator
--	→ Decrement operator

5. Write programs to use all the data types and given arithmetic operations.

6. Write program to perform all the arithmetic operations given in the table.

Deadline: Wednesday Midnight Latest