FAST

National University of Computer and Emerging Sciences Peshawar

Lecture # 03

Software Construction and Development (Java Programming)

Instructor: Muhammad Abdullah Orakzai

DEPARTMENT OF COMPUTER SCIENCE

الذی علم بالقلم۔ علم الانسان ما لم یعلم۔

Operators in Java

Contents

1) Operators in java
2) Unary operators
3) Binary operators
4) Ternary operators
5) Scanner Class in java
6) Character in java

Operator

Operator is a symbol which is used to perform some operation.

Operators are used to perform operations on variables and values.

Types of operators

1) Unary operators
2) Binary operators
3) Ternary operators

1) Unary Operator

1. Increment (++)
2. Decrement (--)
3. Negation (!)

2) Binary Operator

1. Arithmetic (+, -, *, /, %)
2. Relational (>, <, >=, <=, !=, ==)
3. Logical (&&, ||)
4. Assignment (=)
5. Arithmetic Assignment operator (+=, -=, *=, /=, %=)

3) Ternary Operator

- Conditional operator (?:)

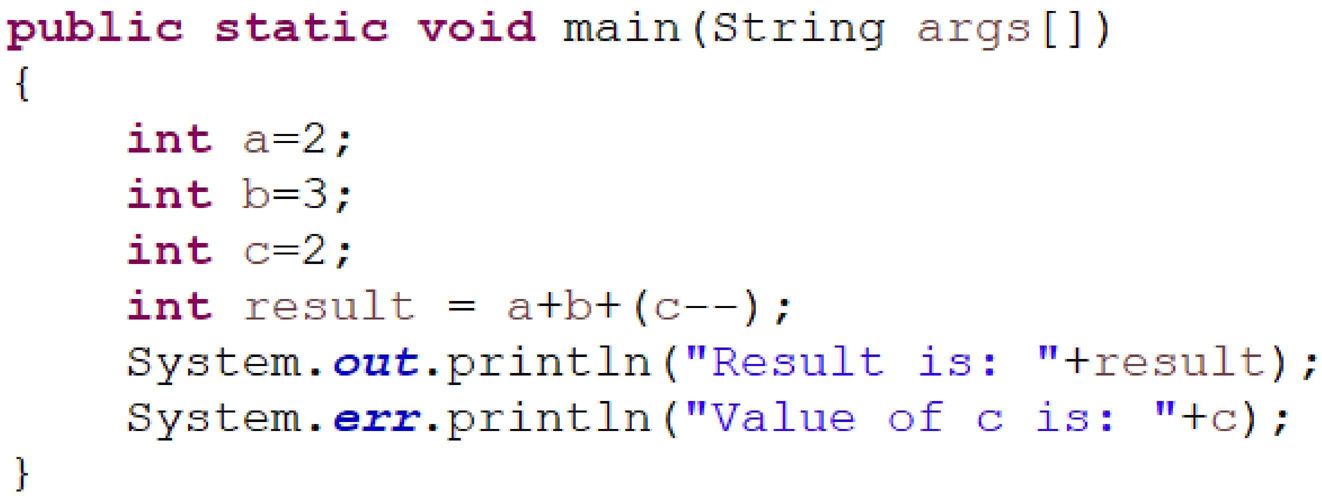
Example

(condition)   **?** statement 1 **:** statement 2;

int result= (n1>n2) ? n1 : n2;

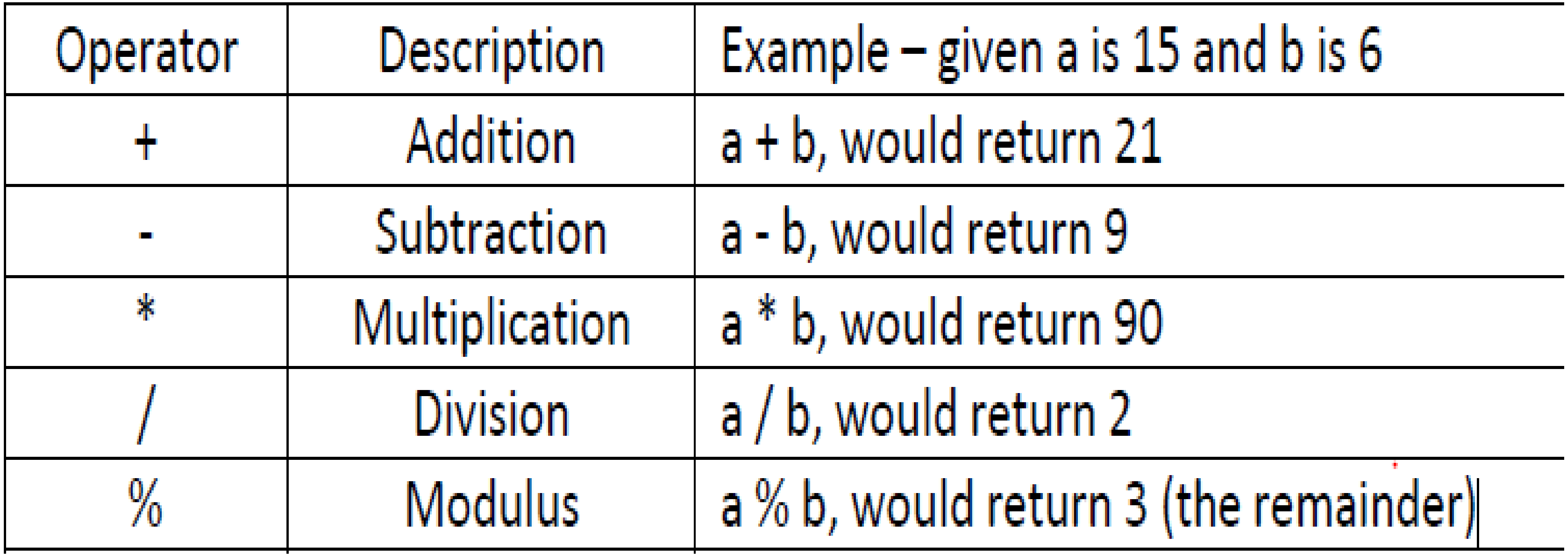
Hard coded values

- Values given during variable declaration.
- Value given to variable in source code.
- **Examples**



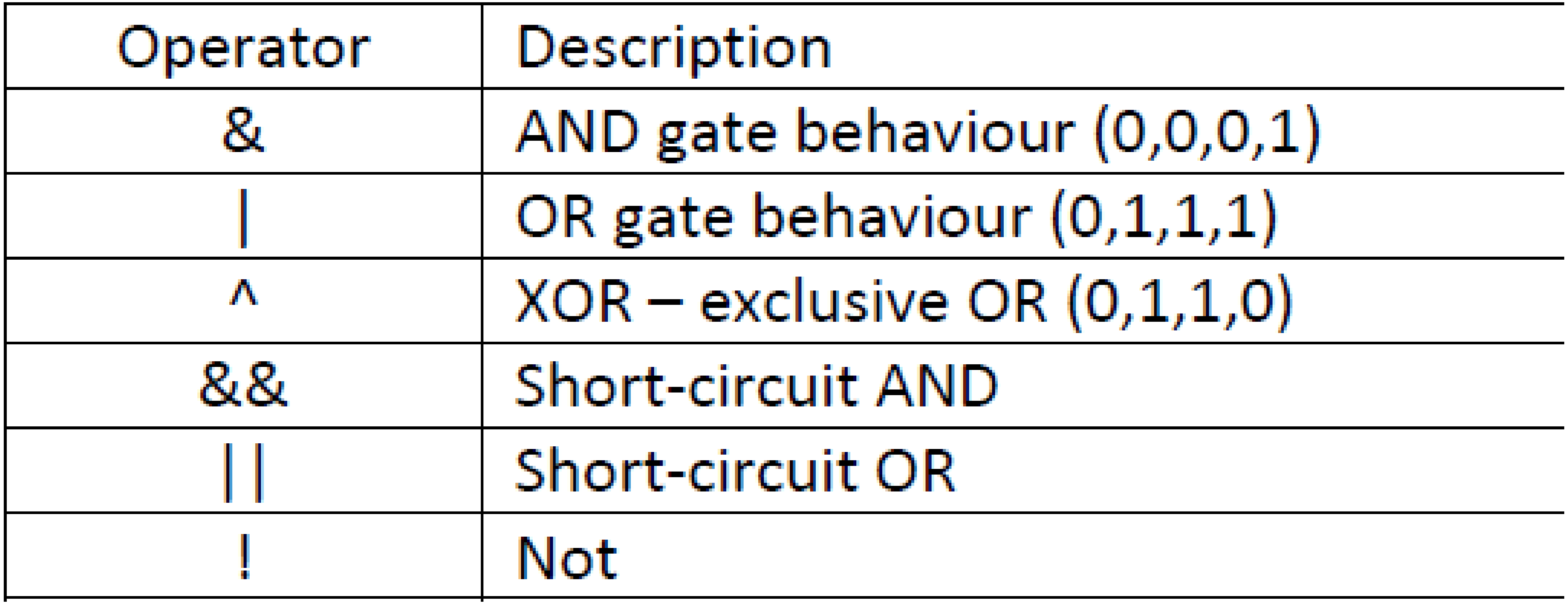
```
public static void main(String args[])
{
    int a=2;
    int b=3;
    int c=2;
    int result = a+b+(c--);
    System.out.println("Result is: "+result);
    System.err.println("Value of c is: "+c);
}
```

Mathematical Operators in Java



Operator	Description	Example – given a is 15 and b is 6
+	Addition	a + b, would return 21
-	Subtraction	a - b, would return 9
*	Multiplication	a * b, would return 90
/	Division	a / b, would return 2
%	Modulus	a % b, would return 3 (the remainder)

Logical Operators in Java



Operator	Description
&	AND gate behaviour (0,0,0,1)
|	OR gate behaviour (0,1,1,1)
^	XOR – exclusive OR (0,1,1,0)
&&	Short-circuit AND
||	Short-circuit OR
!	Not

Increment and Decrement Operators

- The operators that is used to add 1 to the value of a variable is called increment operator.

- The operator that is used to subtract 1 from the value of a variable is called decrement operator.

The Increment Operator (++)

- The increment operator is represented by a double plus (++) sign.
- It is used to add 1 to the value of an integer variable.
- This variable can be used before or after the variable name.
- For example, to add 1 to a value of variable xy, it is normally written as

```
xy = xy + 1;
```

- By using increment operator “++” it is written as

```
xy++
```

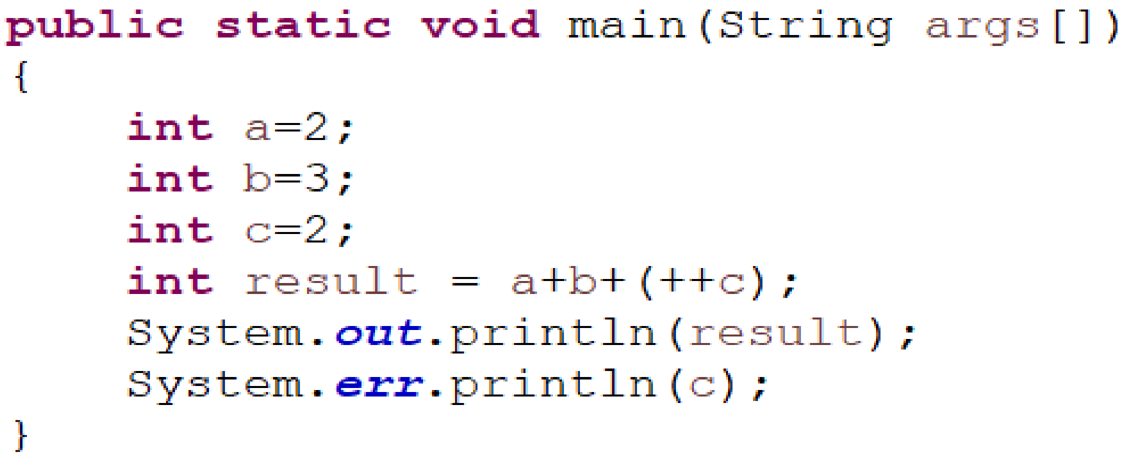
The Increment Operator (++)…

- The increment operator can be written either before or after the variable.
- If it is written before the variable, it is known as prefixing.
- If it is written after the variable, it is known as post fixing.
- Prefix and postfix operators have different effects when they are used in expressions.

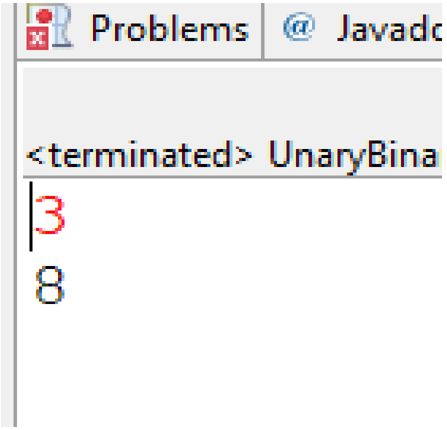
Prefix Increment Operator

- When an increment operator is used in prefix mode in an expression, it adds 1 to the value of the variable **before** the values of the variable is used in the expression.

- For Example



```
public static void main(String args[])
{
        int a=2;
        int b=3;
        int c=2;
        int result = a+b+(++c);
        System.out.println(result);
        System.err.println(c);
}
```



<terminated> UnaryBina

```
3
8
```

Prefix Increment Operator…

- In the above program, 1 will be added to the value of **c** before it is used in the expression.
- Thus after execution, the result will be equal to 8 and the value of **c** will be 3.

Postfix Increment Operator

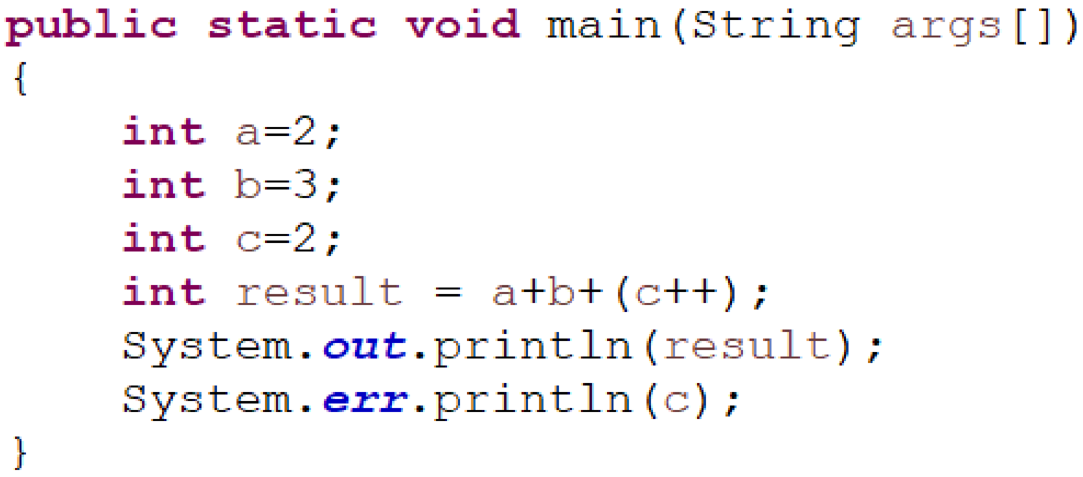
- When an increment operator is used in postfix mode in an expression, it adds 1 to the value of the variable **after** the values of the variable is used in the expression.

- For Example, if in the above example, increment operator is used in postfix mode, the result will be different. The statement will be shown below:

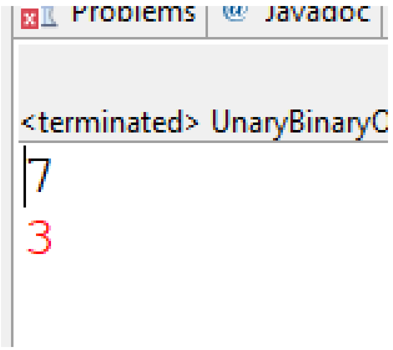
```
result =a + b + c++;
```

Postfix Increment Operator…

- In this case, 1 will be added to the value of c after its existing value has been used in the expression. Thus after execution, the result will be equal to 7 and the value of c will be 3.



```
public static void main(String args[])
{
    int a=2;
    int b=3;
    int c=2;
    int result = a+b+(c++);
    System.out.println(result);
    System.err.println(c);
}
```



```
<terminated> UnaryBinaryC
7
3
```

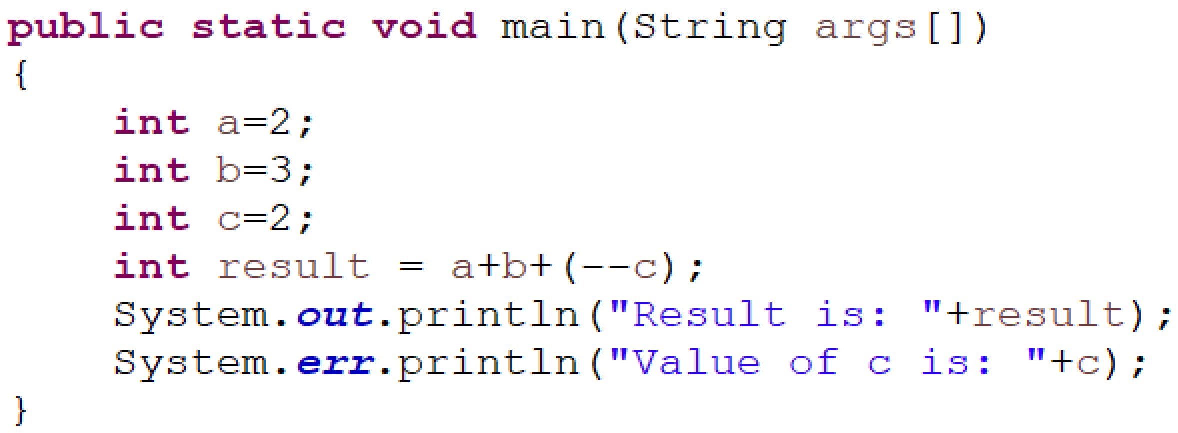
The Decrement Operator (--)

- The decrement operator is represented by a double plus (--) sign.
- It is used to subtract 1 from the value of an integer variable.
- This variable can be used before or after the variable name.
- For example, to subtract 1 from the value of variable xy, the decrement statement is written as:

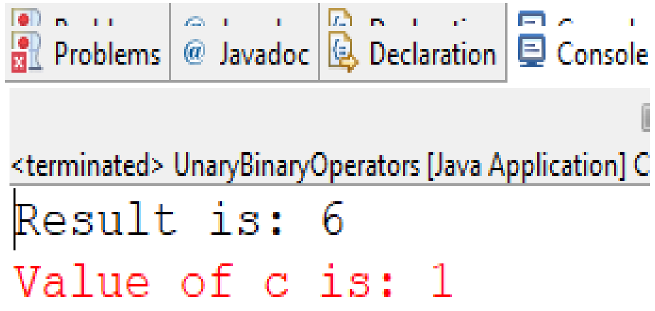
```
xy--;      or    --xy;
```

Prefix Decrement Operator

- When decrement operator is used in prefix mode in an expression, it subtracts 1 from the value of the variable **before** the values of the variable is used in the expression.
- For Example



```
public static void main(String args[])
{
     int a=2;
     int b=3;
     int c=2;
     int result = a+b+(--c);
     System.out.println("Result is: "+result);
     System.err.println("Value of c is: "+c);
}
```



<terminated> UnaryBinaryOperators [Java Application] C

```
Result is: 6
Value of c is: 1
```

Prefix Decrement Operator…

- In the above program, 1 will be subtracted from the value of **c** before it is used in the expression.
- Thus after execution, the result will be equal to 6 and the value of **c** will be 1.

Postfix Decrement Operator

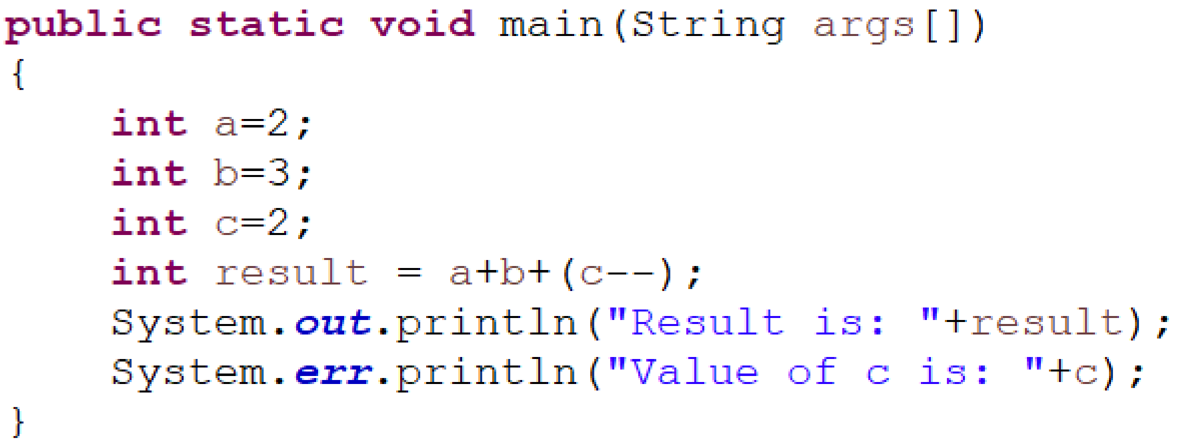
- When an decrement operator is used in postfix mode in an expression, it subtracts 1 from the value of the variable **after** the values of the variable is used in the expression.

- For Example, if in the above example, decrement operator is used in postfix mode, the result will be different. The statement will be shown below:

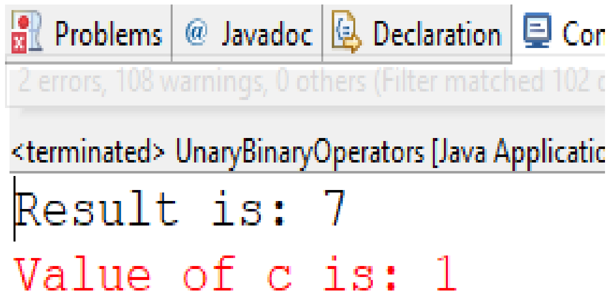
```
result =a + b + c--;
```

Postfix Decrement Operator…

- In this case, 1 will be subtracted from the value of **c** after its existing value has been used in the expression. Thus after execution, the result will be equal to 7 and the value of c will be 1.



```
public static void main(String args[])
{
     int a=2;
     int b=3;
     int c=2;
     int result = a+b+(c--);
     System.out.println("Result is: "+result);
     System.err.println("Value of c is: "+c);
}
```



```
2 errors, 108 warnings, 0 others (Filter matched 102 c
<terminated> UnaryBinaryOperators [Java Applicatic
Result is: 7
Value of c is: 1
```

Runtime values

- Values given to variable during program execution.

Scanner Class

Scanner is built in class which is used to get data from users.

Method in Scanner Class

1) **nextInt()** To get integer values
2) **nextFloat()** To get float values
3) **nextDouble()** To get double values
4) **next()** To get String values. It will not accept space
5) **nextLine()** To get String values. It will accept space

Scanner Class object

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

Note: Scanner class is contained in “ java.util ” pakage of java.

Import Statement:

```
Import java.util.Scanner;
```

String Class

String is predefined class. String is treated as object in java.

Syntax:

```
String variableName;
```

Example

```
String name, fName;
System.out.println(“Enter name”);
name= input.nextLine();
```

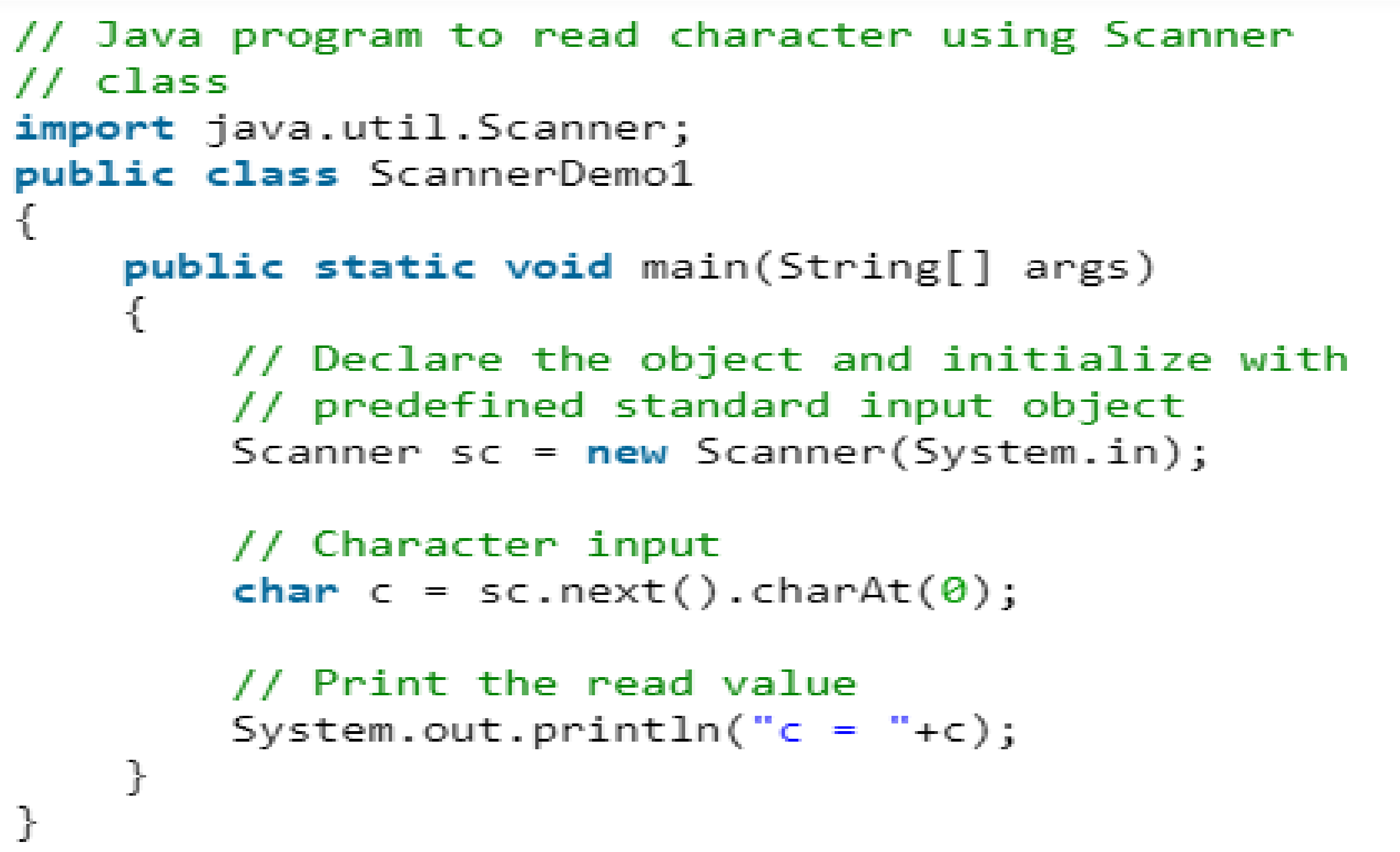
Character in Java

Scanner class in Java supports nextInt(), nextLong(), nextDouble() etc. **But there is no nextChar()**

To read a char, we use **next().charAt(0)**. next() function returns the next token/word in the input as a string and **charAt(0)** function returns the first character in that string.

char letterA = 'A’;

Character in Java…

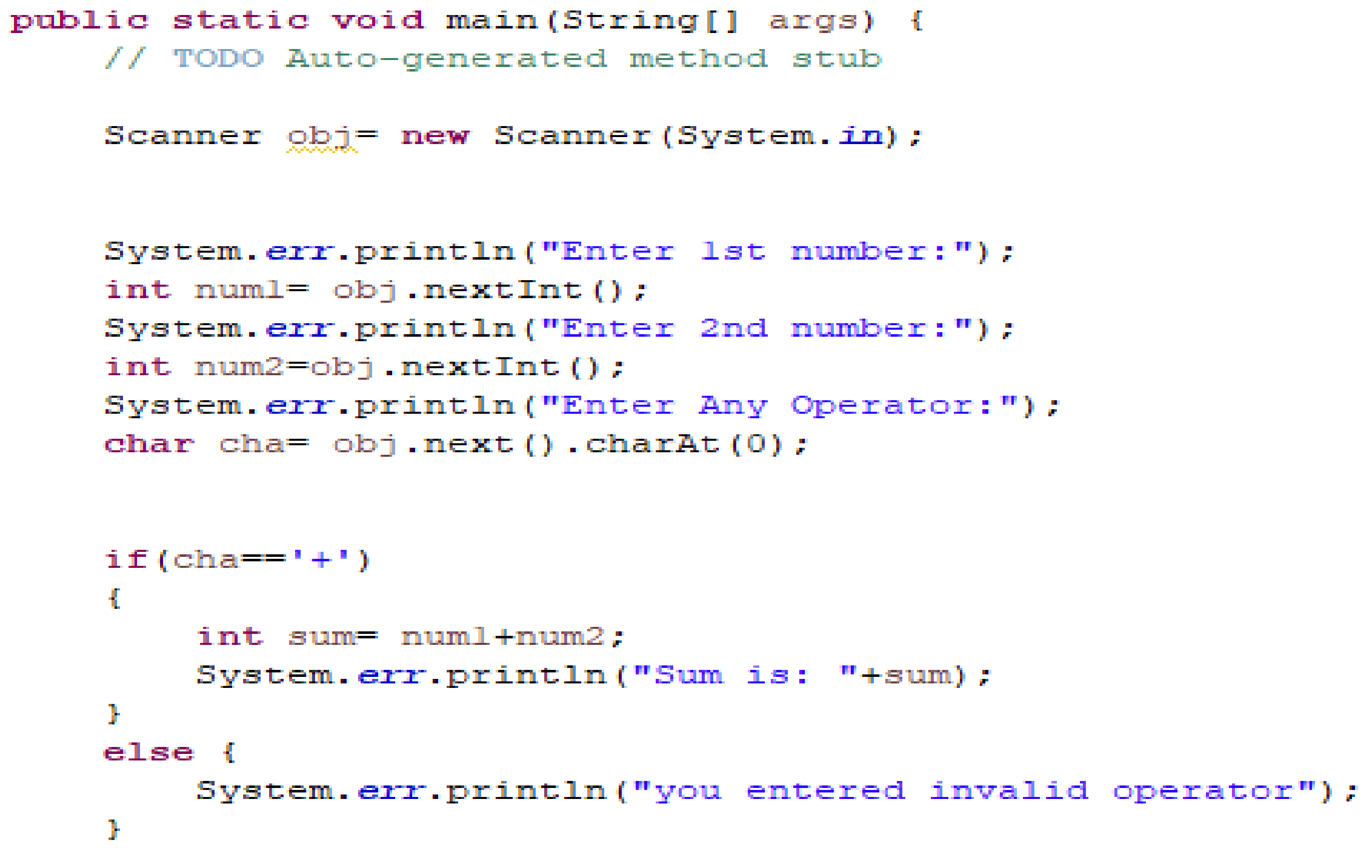


```java
// Java program to read character using Scanner
// class
import java.util.Scanner;
public class ScannerDemo1
{
    public static void main(String[] args)
    {
        // Declare the object and initialize with
        // predefined standard input object
        Scanner sc = new Scanner(System.in);

        // Character input
        char c = sc.next().charAt(0);

        // Print the read value
        System.out.println("c = "+c);
    }
}
```

Character in Java…



```
public static void main(String[] args) {
    // TODO Auto-generated method stub

    Scanner obj= new Scanner(System.in);


    System.err.println("Enter 1st number:");
    int num1= obj.nextInt();
    System.err.println("Enter 2nd number:");
    int num2=obj.nextInt();
    System.err.println("Enter Any Operator:");
    char cha= obj.next().charAt(0);


    if(cha=='+')
    {
        int sum= num1+num2;
        System.err.println("Sum is: "+sum);
    }
    else {
        System.err.println("you entered invalid operator");
    }
```

THANK YOU