Object Oriented Programming

VARIABLE, DATA TYPE, OPERATORS

What is Variable?

A variable is the name of a reserved area allocated in memory

Types of Variables

There are three types of variables in Java:

- Local variable
- Instance variable
- Static variable

Local Variable

- A variable declared inside the body of the method is called local variable.
- A local variable cannot be defined with "static" keyword.

Instance Variable

- A variable declared inside the class but outside the body of the method, is called an instance variable.
- It is called an instance variable because its value is instance-specific and is not shared among instances

Static Variable

- A variable that is declared as static is called a static variable
- Cannot be local
- You can create a single copy of the static variable and share it among all the instances of the class.

Example

```
public class A
{
    static int m=100;//static variable
    void method()
    {
        int n=90;//local variable
    }
    public static void main(String args[])
    {
        int data=50;//instance variable
    }
}//end of class
```

Data Types in Java

- Primitive data types:
  The primitive data types include boolean, char, byte, short, int, long, float and double.

- Non-primitive data types:
  The non-primitive data types include Classes, Interfaces, and Arrays.

Primitive Data Types

- Java defines eight simple types:
  1. byte – 8-bit integer type
  2. short – 16-bit integer type
  3. int – 32-bit integer type
  4. long – 64-bit integer type
  5. float – 32-bit floating-point type
  6. double – 64-bit floating-point type
  7. char – symbols in a character set
  8. boolean – logical values true and false

Primitive Data Types

Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Operator

Operator in Java is a symbol that is used to perform operations.

For example: +, -, *, / etc.

Types of Operators

There are many types of operators in Java which are given below:

- Unary Operator
- Arithmetic Operator
- Shift Operator
- Relational Operator
- Bitwise Operator
- Logical Operator
- Ternary Operator
- Assignment Operator

Operator Type	Category	Precedence
Unary	postfix	*expr*++ *expr*--
	prefix	++*expr* --*expr* +*expr* -*expr* ~ !
Arithmetic	multiplicative	* / %
	additive	+ -
Shift	shift	<< >> >>>
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	logical AND	&&
	logical OR	||
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

Java Unary Operator Example: ++ and --

```
public class OperatorExample{
public static void main(String args[]){
int x=10;
System.out.println(x++);//10 (11)
System.out.println(++x);//12
System.out.println(x--);//12 (11)
System.out.println(--x);//10
}}
```

Java Unary Operator Example: ++ and --

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=10;
System.out.println(a++ + ++a);
System.out.println(b++ + b++);
}}
```

Java Unary Operator Example: ++ and --

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=10;
System.out.println(a++ + ++a);//10+12=22
System.out.println(b++ + b++);//10+11=21
}}
```

Java Unary Operator Example: ~ and !

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=-10;
boolean c=true;
boolean d=false;
System.out.println(~a);//-11 (minus of total positive value which starts from 0)
System.out.println(~b);//9 (positive of total minus, positive starts from 0)
System.out.println(!c);//false (opposite of boolean value)
System.out.println(!d);//true
}}
```

Java Arithmetic Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=5;
System.out.println(a+b);//15
System.out.println(a-b);//5
System.out.println(a*b);//50
System.out.println(a/b);//2
System.out.println(a%b);//0
}}
```

Java Left Shift Operator

```
public class OperatorExample{
public static void main(String args[]){
System.out.println(10<<2);//10*2^2=10*4=40
System.out.println(10<<3);//10*2^3=10*8=80
System.out.println(20<<2);//20*2^2=20*4=80
System.out.println(15<<4);//15*2^4=15*16=240
}}
```

Java Right Shift Operator

```
public OperatorExample{
public static void main(String args[]){
System.out.println(10>>2);//10/2^2=10/4=2
System.out.println(20>>2);//20/2^2=20/4=5
System.out.println(20>>3);//20/2^3=20/8=2
}}
```

Java AND Operator Example: Logical && and Bitwise &

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=5;
int c=20;
System.out.println(a<b&&a++<c);//false && true = false
System.out.println(a);//10 because second condition is not checked
System.out.println(a<b&a++<c);//false && true = false
System.out.println(a);//11 because second condition is checked
}}
```

Java Assignment Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=20;
a+=4;//a=a+4 (a=10+4)
b-=4;//b=b-4 (b=20-4)
System.out.println(a);
System.out.println(b);
}}
```

Java Ternary Operator

Java Ternary operator is used as one line replacement for if-then-else statement and used a lot in Java programming

Java Ternary Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=2;
int b=5;
int min=(a<b)?a:b;
System.out.println(min);
}}
```

If the condition is true then the **first statement** will be selected
If not, then the **second statement** will be selected

Java Ternary Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=7;
int min=(a==b)?a:b;
System.out.println(min);  //min?
}}
```

Java Ternary Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=7;
int min=(a==b)?++a:b++;
System.out.println(min);  //min?
}}
```

Java Ternary Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=7;
int min=(a==b)?++a:b++;
System.out.println(min);  //min=7
}}
```

Java Ternary Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=7;
int min=(a!=b)?++a:b++;
System.out.println(min);  //min?
}}
```

Java Ternary Operator

```
public class OperatorExample{
public static void main(String args[]){
int a=10;
int b=7;
int min=(a!=b)?++a:b++;
System.out.println(min);  //min=11
}}
```