Variables:

Named data storage

Strongly Typed

Variable Naming:

Use only letters and Numbers

First character cannot be number

Style Names using Camel case

Start each word after the first upper case

All other letters are lower cases

public class Varibales {  
 public static void main(String[] args) {  
  
 int myVal=90;  
 System.*out*.println(myVal);  
  
 }  
}

Variables can be declared as final:

Use final modifier

Value cannot be changed once set

Compilation error below with final as we are reassigning :

public class Varibales {  
 public static void main(String[] args) {  
  
 final int myVal=90;  
 myVal=100;  
 System.*out*.println(myVal);  
  
 }  
}

Primitive data types:

**Int type:**

**byte 8**

**shot 16**

**int 32**

**long 64**

public class Intvariables {  
  
 public static void main(String[] args) {  
 int intValue =300000000;  
 long longValue =3000000008000000000L;  
 short shortValue =1277;  
 byte byteValue =127;  
 System.*out*.println("int value "+ intValue);  
 System.*out*.println("long value "+ longValue);  
 System.*out*.println("short value "+ shortValue);  
 System.*out*.println("byte value "+ byteValue);  
 }  
}

float type:

**float**

**double**

**char**

**boolean**

**Ex:**

public static void main(String[] args) {  
 float kmmarothon=42.19f;  
 float ce=-374.98f;  
 System.*out*.println(kmmarothon);  
 System.*out*.println(ce);  
 double salary=300000.00d;  
 System.*out*.println(salary);  
}

Boolean

**Booean:**

public class Boolean {  
  
 public static void main(String[] args) {  
  
 boolean iseligbleforPromotion=true;  
 System.*out*.println(iseligbleforPromotion);  
 }  
}

Char type:

**Char:**

public class CharVaribles {  
  
 public static void main(String[] args) {  
 /\*  
 Stores a single unicode character  
 Literal values placed between single quotes  
 \*/  
 final char gender='M';  
 //gender='F';  
 System.*out*.println("Gender is "+ gender);  
 }  
  
}

Athematic Operators:

public class ArthimeticOperations {  
 public static void main(String[] args) {  
 int firstnumber = 40;  
 int secondnumber = 30;  
 int add = firstnumber + secondnumber;  
 int sub = firstnumber - secondnumber;  
 int mul = firstnumber \* secondnumber;  
 int div = firstnumber/secondnumber;  
 int mod = firstnumber %secondnumber;  
  
  
 System.*out*.println("addition of two numbers "+ add);  
 System.*out*.println("sub of two numbers "+ sub);  
 System.*out*.println("mul of two numbers "+ mul);  
 System.*out*.println("div of two numbers "+ div);  
 System.*out*.println("mod of two numbers "+ mod);  
 }  
}

Compound assignment operator:

* ,+= , -= ,/= ,%=

public class CompondAssigmentOperators {  
  
 public static void main(String[] args) {  
  
  
// int myValue=50;  
// myValue+=5;  
  
 int myOtherValue=100;  
 int val1=5;  
 int val2=10;  
 myOtherValue/=val1\*val2;  
 System.*out*.println("myval is "+ myOtherValue);  
  
 }  
  
  
}

Please complete other samples:

Prefix and Postfix:

public class PrefixandPostfix {  
 public static void main(String[] args) {  
 /\*  
  
 ++variable(pre increment operator) vs variable++(post increment operator)  
  
 \*/  
  
 int someValue=90;  
 System.*out*.println(++someValue);  
 System.*out*.println(someValue);  
  
 System.*out*.println("===================================");  
 System.*out*.println("===================================");  
 System.*out*.println("===================================");  
  
  
 int othervalue=90;  
 System.*out*.println(othervalue++);  
 System.*out*.println(othervalue);  
 }  
}

Operator precedence:

package com.org.varibles;  
  
public class OperatorPreceding {  
 /\*  
  
 \*/  
 public static void main(String[] args) {  
 int valA=21;  
 int valB=6;  
 int valC=3;  
 int valD=1;  
  
 int res1=valA-valB/valC;  
 //int res1=21-6/3  
 //21-2=19  
 int res2=(valA-valB)/valC;  
 //res2=(21-6)/3  
 //5  
 System.*out*.println(res1);  
 System.*out*.println(res2);  
  
 int res3= valA/valC \* valD+valB;  
 //res3= 21/3 \* 1+6  
 //res3= 7 \* 1 + 6  
 //res3= 7+6  
 //res3 =13  
 int res4=valA/(valC\*(valD+valB));  
  
 //res4= 21/(3\*(6+1))  
 //res4= 21/(3\*7)  
 //res4= 21/21  
 //res4=1  
 System.*out*.println(res3);  
 System.*out*.println(res4);  
 }  
}