Comments are English words, can be used for code documentation.  
  
**Purpose of Comments:**  
a) To make the code Readable   
  
b) To make the code disable form execution  
  
**Comments Syntax in Java:**  
Use // for single line Comment or partial line as comment  
  
Use /\* statements  
----------  
-----------  
----------  
\*/ for multiple lines comment  
  
Or  
In Eclipse  
  
Select statements  
Source menu -> Add Block Comment  
  
Uncomment:  
Select Comment block  
Source menu -> Remove comment block   
-------------------------------------  
**Usage of Comments in Test Automation**  
a) To write Test case headers  
  
b) To write Method headers  
  
c) To explain complex logic  
  
d) To explain resource usage  
------------------------------  
**Example:**  
public class Comments {  
  
    public static void main (String [] args){  
        int a = 10, b, c; // Declaration of Variables  
        // It is a Sample Program  
        System.out.println(a);// 10  
        b = 50;  
        c = 2;  
        /\*if (a > b) {  
            System.out.println("A is a Big Number");  
        }  
        else {  
            System.out.println("B is a Big Number");  
        }\*/  
    }  
}  
---------------------------------------  
**II) Java Data Types**  
   
Data type is a classification of the type of data that a variable or object can hold in Computer programming.  
  
**Example:**  
  
Character, Integer, String, float etc...  
  
Java supports explicit declaration of Data types.  
(We need to specify the data type before declaration of variables.)  
  
**Syntax:**  
  
dataType VariableName  
  
Example:   
  
int a;  
or  
int a = 100;  
-------------------------------------  
**In Java we have two types of Data types.**  
  
i) Primitive Data types  
  
ii) Non -primitive Data types  
------------------------------  
i) Primitive Data types **(8 data types)**   
Integer Data types  
---------------------  
**1) byte (8 bits)**  
Example:  
  
byte a = 10;  
  
**2) short (16 bits)**  
  
Ex:  
  
short b = 10000;  
  
**3) int (32 bits)**  
  
int c = 100000;  
  
**4) long (64 bits)**  
  
Ex:  
  
long d = 100000L  
-----------------------------  
Rational Data types (Numbers with decimal places)  
  
**5) float (32 bits)**  
Ex:  
  
float a = 1.2;  
  
**6) double (64 bits)**  
  
Ex:  
  
double b = 19.234567;  
----------  
Characters  
  
**7) char (16 bits)**  
  
Ex:  
  
char a = 'Z'  
-------------------  
Conditional  
  
**8) Boolean**  
  
Ex:  
  
boolean a = true;

boolean b = false;

--------------------------------  
ii) Non -primitive Data types  
Non -primitive Data types in Java are objects and Arrays.  
  
Ex:  
  
Button = new Button("OK")  
--------------------------------------------  
**III) Java Modifiers**  
   
Modifiers are keywords that we add to those definitions to change their meanings.  
  
Two types of modifiers in Java:  
  
i) Access Modifiers  
  
ii) Non Access Modifiers  
-------------------------------  
i) Access Modifiers  
We use Access modifiers to define Access control for classes, methods and variables.  
  
There are 4 types of Access modifiers in Java  
  
**1) Private**  
  
The private access modifier is accessible only within class.  
  
Ex:  
  
class Abc {  
private int a = 100;  
.  
.  
.  
}  
--------------------------  
**2) default**  
  
If we don't use any modifier then it is treated as default, this can be accessible only within package.  
  
Ex:  
class Sample{  
int a = 10;  
.  
.  
}  
-----------------------------------  
**3) protected**  
  
The protected access modifier is accessible within package and out side of the package but  
through Inheritance only.  
-----------------------------  
**4) public**  
  
public access modifier is accessible everywhere.  
  
Ex:  
  
public class Abc{  
.  
.  
}  
--------------------------------------------------  
Modifier Within class    within package    outside of the package subclass Outside of the package  
----------------------------------------------------------------------------------------  
private      Y        N         N                     N  
------------------------------------------------------------------------------------  
default   Y        Y        N                     N  
-----------------------------------------------------------------------------------  
protected Y        Y         Y                    N  
----------------------------------------------------------------------------------  
public    Y        Y         Y                    Y  
---------------------------------------------------------------------------------  
ii) Non Access modifiers  
   
**1) static**  
  
static modifiers are used to create class variable and class methods which can be accessed without instance of a class.  
  
Ex:  
  
class A {  
static String name = "Selenium";  
}  
---------------------  
**2 final**  
final modifier for finalizing the implementation of classes, methods and variables.  
  
class A {  
int a = 10;  
final int b = 20;  
.  
.  
a = 30;  
b = 50;//Incorrect  
}  
--------------------  
**3) abstract**  
  
abstract modifier is to create abstract classes and abstract methods.  
  
Ex:  
  
abstract sample {  
Statements  
--------  
---------  
------  
}  
------------------  
**4) synchronized**   
--------------------------------  
**IV) Variables in Java**  
   
1) What is Variable?  
A named memory location to store or hold the data.  
  
Two types of memory in computer environment:  
  
i) Primary memory - RAM  
  
ii) Secondary memory -ROM  
  
Ex: CD, DVD, HDD, USB drive etc...  
---------------------  
2) Declaration of Variables  
Syntax:  
  
dataType variableName;  
  
Ex:  
  
int a;  
-----------------  
dataType varaible1, variable2, varaible3;  
  
ex:  
  
int a, b, c;  
----------------------  
dataType variableName = value;  
  
Ex:  
  
int a = 100;  
--------------------------  
3) Assigning values to variables  
i) Initialization  
  
Ex:  
int a;  
a = 100;  
  
ii) Reading  
    using input devices  
    from files (text, excel)  
    from databases  
    from Application objects  
--------------------------------  
4) Types of Variables  
In Java we have 3 types of variables.  
  
i) Instance variables  
A variable that is declared inside the class but outside the method.  
  
ii) Local variables  
A variable that is declared inside the Method.  
  
iii) Static / class variables  
A variable that is declared as static, It cannot be local.  
------------------------------  
5) Naming Restrictions  
  
i) Java variables are case sensitive, monkey is not the as MONKEY or Monkey.  
  
ii) Java variable name must start with a letter or $ or \_  
Embedded periods can't be used.  
  
Example:  
  
myvar  
MYVAR   
$myvar   
\_myvar  
myvar1  
myvar\_1  
-----------  
my var  
my.var  
1myvar  
\*myvar  
my-var  
-----------------------  
iii) Variable names cannot be equal to Java reserved words.  
  
Ex:  
int  
for   
import  
-------------------  
iv) Must be unique in the scope of declaration.  
--------------------------------------------  
**Variables example:**  
  
public static void main (String [] args){  
        // Variable Declaration  
        int a;  
        a = 10; // Initialization  
        // Declaration of multiple variables in a statement  
        int b, c, d;  
        b = 20;  
        c = 30;  
        d = 40;  
        // Declaration of multiple variable and Assigning values.  
        int e = 50, f=60, g = 70;  
        char x ='A';  
        double y = 2.345;  
        String z = "Selenium123";  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
        System.out.println(d);  
        System.out.println(e);  
        System.out.println(f);  
        System.out.println(g);  
        System.out.println(x);  
        System.out.println(y);  
        System.out.println(z);  
    }  
}  
-----------------------------------------------  
**V) Java Operators**  
   
Operators are used to perform mathematical, Comparison and Logical operations.  
  
Important categories of Operators:  
  
i) Assignment Operators  
  
ii) Arithmetic operators  
  
iii) Relational operators  
  
iv) Logical Operators  
etc...  
-------------------------------------  
i) Assignment Operators  
1) Assignment operator =  
  
a = 10;  
  
2) Add and Assign Operator +=  
  
a = 10;  
  
a += 20; a=a+20  
  
3) Subtract and Assign -=  
  
a = 10;  
  
a -= 5; equal to a=a-5  
  
4) Multiply and Assign \*=  
  
a = 10  
  
a \*= 5;  
--------------------------------------  
**Example:**  
public static void main (String [] args){  
    int a = 10;  
    System.out.println(a);// 10  
    a += 10;  
    System.out.println(a);//20  
    a -= 10;  
    System.out.println(a);//10  
    a \*= 5;  
    System.out.println(a);//50  
}  
}  
--------------------------------------  
ii) Arithmetic Operators  
Arithmetic Operators return value based Result.  
  
1) Addition +  (for Addition and String concatenation)  
  
2) Subtraction - (for Subtraction and negation)  
  
3) Multiplication \*  
  
4) Division /  
  
5) Modules %  
  
6) Increment ++  
  
7) Decrement --  
-------------------------------  
**Example:**  
  
public static void main (String [] args){  
    int a = 10, b = 5;  
    String c = "Selenium", d = " Testing";  
      
System.out.println("Addition of a, b is: "  + (a + b)); // Addition of a, b is: 15  
System.out.println("Concatenation of c, d is: " + (c+d)); //Concatenation of c, d is: Selenium Testing       
  
System.out.println("Subtraction of a, b is: " + (a-b)); // 5  
System.out.println("Multiplication of a, b is: " + (a\*b)); // 50  
System.out.println("Divison of a, b is: " + (a/b)); //2  
System.out.println("Modulas of a, b is: " + (a%b)); //0  
  
a = ++b;  
System.out.println(a); //6  
b = 5;  
a = --b;  
System.out.println(a);//4  
b =5;  
a = b+4;  
System.out.println(a); //9  
b = 5;  
a = b-4;  
System.out.println(a);//1  
}  
}  
---------------------------------------  
iii) Relational Operators  
Relational operators return boolean or logical result (true or false)  
  
1) ==  
  
2) !=  
  
3) >  
  
4) >=  
  
5) <  
  
6) <=  
------------------------------  
Example:  
public class OperatorsExample {  
public static void main (String [] args){  
    int a = 10, b = 20;  
    System.out.println("a > b is " + (a>b)); //False  
    System.out.println("a >= b is " + (a >= b)); //False  
      
    System.out.println("a < b is " + (a<b)); //True  
    System.out.println("a <= b is " + (a <= b)); //True  
      
    System.out.println("a == b is " + (a == b)); //False  
    System.out.println("a != b is " + (a != b)); //True  
}  
}  
------------------------------------------  
iv) Logical Operators  
1) Logical Not operator !  
  
2) Logical And Operator &&  
  
3) Logical Or operator ||  
--------------------------------  
**Result Criteria for Not operator**  
Operand 1    Operand 2    Result  
----------------------------------------  
true        true        false  
true        false        true  
false        true        true  
false        false        true  
-----------------------------------------  
**Result Criteria for And operator**  
Operand 1    Operand 2    Result  
----------------------------------------  
true        true        true  
true        false        false      
false        true        false          
false        false        false          
-----------------------------------------  
**Result Criteria for Or operator**  
Operand 1    Operand 2    Result  
----------------------------------------  
true        true        true  
true        false        true          
false        true        true              
false        false        false                  
-----------------------------------------  
**Example 1:**  
public class OperatorsExample {  
public static void main (String [] args){  
    boolean a = true, b = false;  
    System.out.println(! (a && b)); // true  
    System.out.println((a && b)); //false  
    System.out.println((a || b));//true  
}  
}  
----------------------------------  
**Example 2:**  
public class OperatorsExample {  
public static void main (String [] args){  
    int a = 100, b = 500, c = 700;  
    if ((a > b) && (a > c)) {  
        System.out.println("A is a Big Number");  
    }  
    else  
    {  
    System.out.println("A is Not a Big Number");  
    }  
}  
}  
-----------------------------------------  
public static void main (String [] args){  
    int a = 100, b = 500, c = 700;  
    if (!(a > b) && !(a > c))  {  
        System.out.println("A is a Big Number");  
    }  
    else  
    {  
    System.out.println("A is Not a Big Number");  
    }  
}  
---------------------------  
public static void main (String [] args){  
    int a = 100, b = 50;  
    if (! (a > b))  {  
    System.out.println("A is a Big Number");  
    }  
    else  
    {  
    System.out.println("B is a Big Number");  
    }  
}  
------------------------------------------------  
Java Flow Control statements  
    Conditional statements  
    Loop statements  
-------------------------------------  
**VI) Java Conditional Statements**  
   
**a) Usage of Conditional statements in Test Automation:**  
i) To insert verification points  
  
ii) Error handling  
--------------  
**b) Two types of conditional statements in Java**  
i) if statement  
  
ii) Switch statement  
-----------------------------  
**c) Types of Conditions**  
i) Single condition  
  
Ex:  
  
if (a > b) {  
Statements  
-------  
--------  
}  
  
  
ii) Compound Condition  
  
Ex:  
  
if ((a > b) && (a < c)) {  
Statements  
-------  
--------  
}  
  
  
iii) Nested condition  
  
Ex:  
  
if (a > b) {}  
 if (a > c) {}  
  if (a > d) {  
  Statements  
  ---------  
  ----------  
}  
-------------------------------------  
d) Usage of Conditional Statements

Usage of Conditional Statements  
1) Executing a block of statements when condition is true.  
**Syntax:**  
if (Condition) {  
Statements  
---------  
---------  
}  
  
**Example:**public static void main (String [] args){  
        int a, b;  
        a =10;  
        b =5;  
          
        if (a > b) {  
            System.out.println("A is a Big Number");  
        }  
------------------------------------  
2) Executing a block of statements when compound condition is true  
**Syntax:**  
if ((condition) && (condition2)) {  
Statements  
----------  
-----------  
}  
  
**Example:**int a, b, c = 2;  
        a =10;  
        b =5;  
          
        if ((a > b) && (a > c)) {  
            System.out.println("A is a Big Nuber");  
        }  
-------------------------------------------  
3) Executing a block of statements when condition is true. otherwise executing another block of statements.  
**Syntax:**  
if (Condition) {  
Statements  
----------  
--------  
}  
else {  
statements  
---------  
--------  
}  
  
**Example:**  
public static void main (String [] args){  
        int a, b;  
        a =10;  
        b =50;  
          
        if (a > b) {  
            System.out.println("A is a Big Number");  
        }  
        else {  
            System.out.println("B is a Big Number");  
        }  
-----------------------------------------  
4) Decide among several Alternates (else if structure)  
**Syntax:**  
if (condition) {  
Statements  
------  
---------  
}  
else if (condition) {  
Statements  
------  
---------  
}  
else if (condition) {  
Statements  
------  
---------  
}  
else  
{  
statements  
----------  
--------  
}  
  
**Example:**  
public static void main (String [] args){  
        int a = -100;  
                  
        if ((a >= 1) && (a < 100)) {  
            System.out.println("A is a Small Number");  
        }  
        else if ((a > 100) && (a <= 1000)) {  
            System.out.println("A is a Medium Number");  
            }  
        else if (a > 1000) {  
            System.out.println("A is a Big Number");  
            }  
        else  
        {  
            System.out.println("A is either Zero or negative value");  
        }  
--------------------------------------------          
5) Executing a block of statements when more than one condition is true (Nested if).  
**Syntax:**  
if (condition) {}  
 if (condition) {}  
  if (condition) {  
Statements  
--------  
---------  
}  
else  
{  
Statements  
---------  
---------  
}  
  
**Example:**  
public static void main (String [] args){  
        int a =10, b =7, c = 5, d = 13;  
                  
        if (a > b){}  
          if (a > c) {}  
           if (a > d){  
               System.out.println("A is a Big Number");  
           }  
           else {  
               System.out.println("A is Not a Big Number");  
           }  
---------------------------------------  
6) Decide among several alternates (using Switch case structure).  
**Syntax:**  
switch (expression) {  
  
case value:  
Statements  
-------  
break;  
case value:  
Statements  
-------  
break;      
case value:  
Statements  
-------  
break;  
default  
Statements  
-------  
------  
}  
  
**Example:**char grade = 'X';  
          
        switch (grade){  
          
        case 'A':  
            System.out.println ("Excellent");  
            break;  
        case 'B':  
            System.out.println ("Well Done");  
            break;  
              
        case 'C':  
            System.out.println ("Better");  
            break;  
        default:  
            System.out.println("Invalid Grade");  
                }