**Java Program Structure and Java Syntax**  
  
i) Java Program Structure  
  
ii) A Sample Java Program  
  
iii) Comments in Java  
  
iv) Java Data Types  
-----------------------------------------  
Java Environment Setup  
   
> Download Java (JDK) Software and Install  
  
> Set Environment Variable (Path variable)  
  
> Download Eclipse IDE and Extract.  
--------------------------  
Create Java project  
    > Create Java Package  
        Create Java Class /Program  
-----------------------------------------  
**i) Java Program Structure**  
   
1) Documentation Section  
  
2) Package declaration Statement  
  
Ex: package xyza;  
  
3) Import Statements  
  
We import built in and User defined libraries using import keyword  
  
Ex:   
  
import java.io.Console;  
import java.lang.\*;  
  
import - It is a Java keyword to import Libraries.  
  
java -Project  
  
io -Package  
  
Console - Class  
  
lang.\*; - import all classes from lang package.  
--------------------------------------  
4) Class declaration Statement  
  
public class Sample {  
  
}  
  
public - Access Modifier  
  
class - Java keyword to declare a class  
  
Sample - it is the Class name (You can use any meaningful name)  
----------------------------------------------  
5) main Method (Java Program execution starts from main method)  
(\* It is the mandatory statement in every Java program)  
  
public static void main (String [] args) {  
}  
  
public - Access Modifier  
  
static - Non-Access Modifier (use main method without invoking any object)  
  
void - Returns nothing  
  
main- method name  
-----------------------------------  
6) Declarations  
  
We can Declare Variables and Constants.  
  
Other Statements  
  
System.out.println("Hello Selenium");  
  
System - Class (Pre-defined)  
  
out - Object  
  
println - method  
  
"Hello Selenium" - Message  
--------------------------------  
7) Code blocks  
  
Condition blocks  
  
Loop blocks  
  
Method blocks (method declaration before the main method, but we access methods after main method)  
etc...  
-------------------------------------  
> Every normal statement/step ends with semi colon  
  
> Every code block enclosed with {}  
-----------------------------------------  
**ii) A Sample Java Program**  
   
//Documentation  
  
package xyza;  
  
import java.io.Console;  
import java.lang.\*;  
  
public class Sample {  
  
//Create a Method(User defined)  
public int multiply(c int a, int b, int c){  
int result = a \* b \* c;  
return result;  
}  
  
public static void main (String [] args){  
// This is a sample Program  
  
int a = 10, b=20, c=30; //Variables Declaration      
  
final int money =100;//Constant Declaration  
  
System.out.println("Addition of a, b is " + (a + b));//Addition of a, b is 30  
System.out.println(money);//100  
System.out.println(c);//30  
  
//Condition Block  
if (a > b){  
System.out.println("A is a Big Number");  
}  
else  
{  
System.out.println("B is a Big Nuber");  
}  
  
//Loop block  
  
for (int d=1; d <=10; d++){  
    System.out.println(d);  
}  
//Create Object and access Methods  
Sample obj = new Sample();  
int x = obj.multiply(10, 25, 50);  
System.out.println(x);  
}  
}  
-----------------------------------------  
**iii) Comments in Java**  
   
Comments are English words used for Code documentation.  
  
Purpose of Comments  
  
    a) To make the code Readable  
  
    b) To make the code disable from execution  
------------------------  
Comments Syntax in Java  
  
Use // for Single line comment  
  
Use /\* ......  
...........  
..............  
\*/ for multiple lines comment  
  
Example:  
package xyza;  
  
public class Sample2 {  
  
public static void main (String [] args){  
//This is a Sample Program  
int a, b, c; //Declaration of variables  
a=10; b=20;c=30;  
          
/\*if (a > b){  
System.out.println("A is a Big Number");  
}  
else {  
System.out.println("B is a Big Number");      
}\*/  
  
System.out.println(c);  
}  
}  
------------------------------------  
Usage of Comments in Test Automation  
  
a) To write Test Case header  
  
b) To write Method header  
  
c) To explain complex logic etc...  
-----------------------------------------  
**iv) Java Data Types**  
   
Data Type is a classification of the type of data that variable or Constant or object can hold in computer programming.  
  
Ex: character, integer, float, boolean etc...  
  
Java Supports Explicit Declaration of Data Types.  
(we need to specify the data type before declaring the a Variable or constant etc....)  
  
Syntax:  
  
dataType variableName;  
  
dataType variableName =value;  
  
dataType variable1Name, variable2Name, variable3Name;  
  
Example:  
  
int a;  
  
char b ='A';  
  
int a, b, c;  
-----------------------------  
Two Types of data Types in Java  
  
a) Primitive Data Types  
  
b) Non-primitive Data Types / Reference Data Types  
--------------------------------  
a) Primitive Data Types (8 data types)  
  
i) Integer Types  
----------------  
1) byte (8 bits)  
  
byte a =10;  
  
2) short (16 bits)  
  
short a =10000;  
  
3) integer(32 bits)  
  
int a = 100000;  
  
4) long (64 bits)  
  
long a =100000000000000;  
------------------------  
ii) Relational types (Numbers with decimal places)  
  
5) float (32 bits)  
  
float a = 1.23;  
  
6) double (64 bits)  
  
double a =123.345654322;  
-------------------------------  
iii) Characters  
  
7) character  
  
char a ='Z'  
----------------------------  
iv) Conditional  
  
8) Boolean   
  
boolean a = true;  
------------------------------  
b) Non-primitive Data Types / Reference Data Types  
  
Non-primitive or Reference data types in Java are Objects and Arrays.