Selenium Class 7 - Java Comments, Modifiers and Data Types

i) Java Comments

ii) Java Modifiers

iii) Java Data Types

---

i) Java Comments

Comments are English words used for Code Documentation

Purpose of Comments:

- To make the Code Readable

- To make the code disable the code from Execution

Java Supports single line comment and multiple line comments.

Syntax:

a) Use // for Single line Syntax

b) Use /*.....

.....

.....

.....*/ for multiple line comments

Or

Comment

Select Statements / Steps

Source Menu (In Eclipse IDE) > Add Block Comment

Uncomment

Select Comment Block > Source Menu (In Eclipse IDE) > Remove Block Comment

Example:

```
public static void main(String[] args) {
// It is a Sample Program
int a, b; //Declaration of Variables
a=10; b=20; // Initialization
System.out.println(a+b);//30

/*if (a > b){
System.out.println("A is a Big Number");
}
else
{
System.out.println("B is a Big Number");
}*/

}
}
```

Usage of Comments in Test Automation

a) To write Test case Header

b) To write Method header

c) To explain Complex Logic etc...

---

ii) Modifiers in Java

Two Categories of Modifiers in Java

a) Access Modifiers

b) Non -Access Modifiers

a) Access Modifiers

Access Modifiers can be used to define access control for classes, methods, and variables.

1) public

public access modifier is accessible every where

Example:

```
public class Sample{
.....
}


public int a =10;


public int add (Parameters){
Method body...
}
```

2) private

The private access modifier is accessible only with in class

```
private int b=200;
```

3) default

if we don't specify any modifier then It is treated as default, this can be accessible only within package

Example:

```
class Sample{
.....
}
```

4) protected

The protected modifier is accessible within package, outside of the package but through Inheritance.

Example:

```
protected class Sample{
....
}
```

Modifier	Within Class	Within Package	Outside of the package (by sub class)	Outside of the package
private	Y	N	N	N
default	Y	Y	N	N
Protected	Y	Y	Y	N
public	Y	Y	Y	Y

b) Non -Access Modifiers

1) static

static modifier is used to create classes, methods, and variables.

Example:

```
static int a=10;
static int add (parameters){
.............
.............
}
```

2) final

final modifier for finalizing classes, methods, and variables.

Example:

final int x =100;//Correct

final int y;//Incorrect

y=200;

3) abstract

abstract modifier is used to create abstract classes and abstract methods.

Example:

abstract class Sample{

.....

}

abstract int add(); //abstract method

public int add(){ //concreate method

……..

}

---

iii) Java Data Types

What is Data Type?

Data Type is a classification of the type of data that a Variable or Constant or Method can hold in computer programming

Ex: Character, Number, Number with decimal values, Logical data, String, Date, Currency etc...

Note: Java supports explicit declaration of Data Types.

(We need to specify the Data Type before declaring Variables, Constants, and methods...)

Syntax:

a) Variables

dataType variableName;

Or

dataType variableName = value;

Or

dataType variable1Name, variable2Name, variable3Name;

or

dataType variable1Name=value, variable2Name= value, variable3Name=value;

Example:

int a;

a=10;

int b=100;

int c, d, e;

c=30; d=40; e=50;

int f=60, g=70, h=80;

b) Constants

Note: Variables and Constants are for holding the data, Variables may vary, but constants never change

int a;

a=100;//Correct for Variable

.

.

.

.

a=30; //Correct for Variable

```
final int b;
b=200; //Incorrect
final int c =300;//Correct
.
.
.
c=400; //Incorrect
```

c) Method with Return Value

```
public int add (int a, int b, int c){
.....
.....
.....
return...;
}
```

Two Categories of Data Types

1) Primitive Data Types
2) Non-primitive data Types

1) Primitive Data Types

a) Integer Data Types

i) byte (8 bits)

- Minimum value is -128 (-2^7), Maximum value is 127 (inclusive)(2^7 -1)
- Default value is 0

byte a=10;

ii) short (16 bits)

- Minimum value is -32,768 (-2^15)
- Maximum value is 32,767 (inclusive) (2^15 -1)
- Default value is 0.

short b=1000;

iii) int (32 bits)

- Minimum value is - 2,147,483,648 (-2^31)
- Maximum value is 2,147,483,647(inclusive) (2^31 -1)
- The default value is 0

int c =10000;

int d=12;

int e=1;

iv) long (64 bits)

- Minimum value is -9,223,372,036,854,775,808(-2^63)
- Maximum value is 9,223,372,036,854,775,807 (inclusive)(2^63 -1)
- Default value is 0L

long f =1000000000000000;

b) Relational Data Types (Numbers with Decimal Places)

v) float (32 bits)

float x = 10.23f;

vi) double (64 bits)

double y = 1234.5678765;

c) Character Data Type

vii) character

char z = 'A';

char s = '1';

char r = '*';

d) Conditional Data Type

viii) boolean

boolean k = true/false;

2) Non Primitive data types / Reference data types

Non Primitive data types in Java are Objects (String, Array etc...)

Example:

Sample a = new sample();

.....

123 - Integer Data Type

'Y' -Character

123.34 - float/double

"Selenium Testing" - String

"abc123*&^" - String

"123" - String

1 - Integer

'1' - Character

.....

How to handle Date type data? - Vijay

Converting Data / Data Conversion

Converting data from one type to another...

Assigning Values to Variables

1) Initialization - No Data Conversation

2) Reading – (Data Conversation is required)

Read Input (using Input Devices)

Read data from files

Read data from Application objects

When Data Conversion is required?

Whenever we read data then computer program considers that data as String type data, we need to convert the data in order to perform mathematical operations, Note: Generally we convert String type to Integer Type/ Relational type, we can't convert Alpha bytes to numbers etc...
