**#. Using Primitive Data Types , we cannot store multiple values in one go.**

**#. To Solve this problem, we must group all values to send them as a single unit.**

**#. To Group them as a single unit,we must store all values in continues memory location as Single Unit Platform,**

**and this can be possible by using non-primitive data types.**

**Non-Primitive DataTypes : String, Array, Interface,Class**

**Operators:-**

**#. It is a symbol that is used to perform operations.**

**1. Unary Operator ( ++,--)**

**2. Arithmetic Operator (+,-,\*,/)**

**3. Relational Operator ( ==,!=,<,>,<=,>=)**

**4. Bitwise Operator (&)**

**5. Logical Operator ( && )**

**6. Assignment Operator ( =,+=,-=,\*=,/=)**

**1. Unary Operator ( ++, -- )**

**1. Post and Pre-Increment (++) :- Increment by 1.**

**2. Post and Pre-Decrement (-- ) :- Decrement by 1.**

**#. Unary Operator requires only one operand.**

**Arithmetic Operator :-**

**#. It is used to perform additon , substraction, multiplication,division.**

**Logical Operator ( && )**

**#. The Logical Operator ( && ) does not check the second condition if the first condition is false.**

**Bitwise Operator ( & )**

**#. The Bitwise Operator ( & ) will check both the conditions always whether first condition is true or false.**

**AND**

**TRUE TRUE = TRUE**

**TRUE FALSE = FALSE**

**FALSE TRUE = FALSE**

**FALSE FALSE = FALSE**

**OR**

**TRUE TRUE = TRUE**

**FALSE TRUE = TRUE**

**TRUE FALSE =TRUE**

**FALSE FALSE = FALSE**

**Assignment Operator.**

**#. It is used to assign direct value to any variable using arithmetic Operator.**

**Relational Operator:-**

**#. Java has six relational operator that compare 2 numbers and return boolean value.**

**#. The Relational Operator are ( <,>,<=,>=,==,!=)**

**1. x < y : Less Than :- True if x is less than y, otherwise false.**

**2. x > y : Greater Than :- True if x is greater than y , otherwise false**

**3. x <= y : Less Than or Equal ti :- True if x is less than or equal to y,otherwise false.**

**4. x >= y : Greater Than or Equal to :- True if x is GReater Than or Equal to y, otherwise false.**

**5. x== y : Equal : True if x is equal to x, otherwise false.**

**6. x !=y : Not Equal :- True if x is not equal to y,otherwise false.**

**Conditional Statements:-**

**1. if statement**

**2. if else statement**

**3. if else-if ladder statement**

**4. nested if statement**

**5. switch**

**1. if statement**

**#. it checks the condition**

**#. it will exceute if the condition will be true for if block.**

**Syntax:- if(condition)**

**{**

**// executable code**

**}**

**1. if else statement**

**#. it checks the condition**

**#. it will exceute if the condition will be true for if block. otherwise else block will execute.**

**Syntax:-**

**if(condition)**

**{**

**// executable code**

**}**

**else**

**{**

**// executable code**

**}**

**3. if else-if ladder**

**#. It checks the condition.**

**#. it will execute for multiple conditions.**

**Syntax:-**

**if (condition1 )**

**{**

**// executebale block**

**}**

**else if ( condition2)**

**{**

**// executable code**

**}**

**else {**

**// executable code**

**4. Nested if statement:-**

**#. It is also used to check the condition.**

**#. In case of nested if statement if block will exist inside another if block.**

**Syntax:-**

**if (condition)**

**{**

**// execuatble code**

**if(condition)**

**{**

**// executable code**

**}**

**}**

**}**

**Switch :-**

**#. Instead of using many if else...statements,we can use the switch statement.**

**#. The switch statement select one of many code block to be executed.**

**Syntax:-**

**switch(expression){**

**case x :**

**// execuatbel code**

**break;**

**case y :**

**// execuatble code**

**break;**

**default :**

**// execuatble code**

**}**

**Loop:-**

**#. When we need to execute a block of code multiple times , we use loop.**

**1. for loop**

**2. for each loop**

**3. while loop**

**4. do while**

**Loop:-**

**1. for loop**

**#. it is used to iterate the block of code multiple times.**

**#. if the number of iterations is fixed then it is recommended to use for loop.**

**Sytanx:-**

**for ( initilization;condition;upadtation)**

**{**

**// executable code;**

**}**