**Scala Variable :**

var data = 100

data = 101  // It works, No error.

val data = 100

data = 101  // Error: reassignment to val

## Multiple assignments:

## Will assign Int Data type for value sai and String for value sai2.

## Val(sai:Int,sai2:String)=Pair(23,”Buriburi”)

## Default:

## When you assign an initial value to a variable, the Scala compiler can figure out the type of the variable based on the value assigned to it.

## This is called variable type inference.

## Val(sai,sai2) =Pair(23,”Buriburi”)

Scala Example: If Statement

var teamcount:Int = 20;

if(teamcount = 10){

    println ("The team has enough members")

}

## Scala If-Else Statement

if(condition){

    // If block statements to be executed when the condition is true

} else {

    // Else block statements to be executed when the condition is false

}

## Scala If-Else-If Ladder Statement

## If condition 1 doesn’t satisfy it will go to else if condition 2.

## If the 2nd condition doesn’t satisfy will go t following else-if.

## At last if all of the condition doesn’t satisfy then it will print the else statement.

if (condition1){

//Code to be executed if condition1 is true

} else if (condition2){

//Code to be executed if condition2 is true

} else if (condition3){

//Code to be executed if condition3 is true

}

...

else {

//Code to be executed if all the conditions are false

}

## Scala Pattern Matching Example2

## When enters into main will check for the string “Hello”.

## Will try to find the match in the search class.

## In case 1 it doesn’t match.

## Case 2 doesn’t match so it will switch to case 3.

## But in case three the word matches so it will print the word “Hello”.

object MainObject {

   def main(args: Array[String]) {

        var result = search ("Hello")

        print(result)

    }

    def search (a:String) = a match{

        case 1  => println("One")

        case "Two" => println("Two")

        case "Hello" => println("Hello")

        case \_ => println("No")

           }  }

**BASIC SCALA PROGRAMMING FOR ADDING:**

* xc and yc are assigned as integer.
* When we call class point the values from the main class will be assigned to xc and yc.
* Then those values will be stored into x and y.
* In class move we define dx and dy as integer
* Inside the class we add x+dx and y+dy and store those integers to x and y respectively.
* At last we print the x and y location.
* In object class we move (10,10) to val xc and val yc.
* As well var x is assigned value 10.
* var y is assigned value 20.
* x+dx=10+10=20
* y+dy=10+20=30
* Here class demo is the main class.

import java.io.\_

class Point(val xc: Int, val yc: Int) {

var x: Int = xc

var y: Int = yc

def move(dx: Int, dy: Int) {

x = x + dx

y = y + dy

println ("Point x location : " + x);

println ("Point y location : " + y);

}

}

object Demo {

def main(args: Array[String]) {

val pt = new Point(10, 20);

// Move to a new location

pt.move(10, 10);

}

}

## Calling Functions

* In the main class we call addint class and assign the value sto the addint class from the main class.
* Where a is assigned as 5 and b as 7.
* In next step sum is initiazed as 0.
* And sum=a+b where 5+7
* Return sum and the value will be printed.

object Demo {

def main(args: Array[String]) {

println( "Returned Value : " + addInt(5,7) );

}

def addInt( a:Int, b:Int ) : Int = {

var sum:Int = 0

sum = a + b

return sum

}

}