14-Feb-2020

-----------

ii. Nested if:

---------

To check more than one condition, based on that to execute particular block of statements we can use Nested If

In simple, placing "if statement" inside another if statement is called Nested if

Syntax:

------

if (condition-1) {

--------------

-------------- //statements-1

}

else if (condition-2){

-------------

-------------//statements-2

}

else {

--------------

--------------//statements-3

}

Ex: Write program to find smaller value in given 3 numbers

Script:

------

// To perform Nested if program using 3 values

int a=30;

int b=20;

int c=10;

if ((a<b) && (a<c)){

System.out.println("smaller value is: "+a);

}

else if (b<c) {

System.out.println("smaller value is "+b);

}

else {

System.out.println("smaller value is: "+c);

}

}

}

----------------------------------------------------------------------------------------------------------------------------------------------

Ex: Write program to find student grade based on given marks where as

marks are 90 or above will be Grade "A"

marks are 80 or above will be Grade "B"

marks are 70 or above will be Grade "C"

other than those marks then Grade is"D"

Script :

// To find student grade on given marks

int stMarks;

String stGrade="";

// To read student marks from console\

Scanner sc=new Scanner(System.in);

System.out.println("Enter student marks");

stMarks=sc.nextInt();

if(stMarks>=90){

stGrade="A";

}

else if (stMarks>=80){

stGrade="B";

}

else if (stMarks>=70) {

stGrade="C";

}

else {

stGrade="D";

}

System.out.println("Student marks are:"+stMarks);

System.out.println("Student Grade is:"+stGrade);

}

}

-------------------------------------------------------------------------------------------------------------------------------------------------

Ex: Write program to check person eligible criteria to work based on age between 18 to 60

// To test Age

Scanner sc=new Scanner(System.in);

System.out.println("Please enter your age");

int age=sc.nextInt();

if (age<18) {

System.out.println("You are minor");

System.out.println("Your not eligible to work");

}

else if (age>=18 && age<=61) {

System.out.println("Your are eligible to work");

}

else {

System.out.println("Your too old to work as per Govt rules");

}

}

}

--------------------------------------------------------------------------------------------------------------------------------------------------------

iv. switch statement

It is same like nested if, but it will increase reliability of the script.

Syntax:

switch (expression)

{

case (condition-1):

-------------

------------- //statements-1

break;

case (condition-2):

-------------

-------------// statements-2

break;

case (condition-3)

------------

------------//statements-3

break;

default

--------------

-------------// statement-4

}

=====================================================================================================================================

Ex. Write program to display color name based on given character from "RGB"

Scanner sc= new Scanner(System.in) // [Ctrl+Shift+O]-- to import package

System.out.println("Enter any one character from: RGB");

String myColor=sc.nextLine();

switch (my Color) {

case "R" :

System.out.println("Your choice is: "RED");

break;

case "G" :

System.out.println("Your choice is: GREEN");

break;

case "B":

System.out.println("Your choice is: BLUE");

break;

default:

System.out.println("Invalid entity");

}

======================================================================================================================================

b Loop Statements/ Iterative Statements

-------------------------------------

To execute some portion of the script multiple iterations we can use iterative / loop statements in a program

i. While Loop

ii. Do Loop

iii. For loop

(i) While Loop

----------

It will allow script to execute multiple iterations until given condition is true

Syntax:

while (condition)

{

----------

---------- // Repeatable statements

}

Ex. To print from 1 to 5 using while loop

int n=1;

while (n<=5) {

System.out.println(n);

n++;

}

O/P if n=5 if n=6

O/P: 5 no O/P

-------------------------------------------------------------------------------------------------------------------------------------------

(ii) Do Loop:

-------

It is same as while loop, but it will allow atleast one time to execute block of statements irrespective of condition true/false

Syntax:

do {

----------

---------- // statements

[break] --> optional if it is []

} while (condition);

Note: break ---> It will terminate the iterations, it is optional

\*\*\* Difference between while and do

Do --> will allow to execute the result atleast once

While -> will not display any result or output

--------------------------------------------------------------------------------------------------------------------------------------

Ex.1

int n=1;

do {

System.out.println(n);

n++;

} while (n<=5);

-------------------------------------------------------------------------------------------------------------------------------------

Ex.2

int n=1;

do {

System.out.println(n);

if (n==3) {

break;

}

n++;

} while (n<=5);

----------------------------------------------------------------------------------------------------------------------------------------

Ex-3

int n=6;

do {

System.out.println(n);

n++;

} while (n<=5);

----------------------------------------------------------------------------------------------------------------------------------------

(iii) For loop

To execute block of statements fixed no of iterations we can use For loop

Syntax

for (initialization; condition ; increment/ decrement) {

--------------------

-------------------- // Repeatable statements

[break];

}

Note : " If we know the fixed no of records go for For Loop

If we don't know the limitations/count go for While Loop "

=============================================================================================================================================

Ex-1

for (int i=1; i<=5 ; i++) {

System.out.println(i);

}

------------------------------------------------------

Ex-2

for(int i=1;i<=5;i++){

System.out.println(i);

if(i==3)

break;

}

}

----------------------------------------------------

Ex-3

Write for loop to print from 5 to 1

Script:

for(int i=5;i>=1;i--){

System.out.println(i);

}

-----------------------------------------------------

====================================================================================================================================================

Note:

\*\*\* For-each:

--------

Which is used to read values from an array/collections

Syntax:

------

for(datatype variableName : array/collectionsname){

---------------------

---------------------

}

It will read each value from array for each iteration and those values will be assigned to variable

=====================================================================================================================================================

ARRAY DECLARATION

-----------------

To store multiple values with same data type we can use array

Syntax:

------

datatype[]array\_name= new datatype[size];

size --> number of values to store

array index starts with zero

--> To find number of values in an Array we can use array.length

===================================================================================================================================================

Ex: Create an array to store 3 values

// Array Declaration

String[]myCars= new String[3];

// To assign values

myCars[0]= "Audi";

myCars[1]= "Benz";

myCars[2]= "BMW";

// To find the size of an array

System.out.println("Number of values are: "+myCars.length);

// To read values from array using Index

System.out.println(myCars[0]);

System.out.println(myCars[1]);

System.out.println(myCars[2]);

System.out.println("\*\*\*\*\*\*\*");

// To read values from array using for loop

for(int i=0;i<=myCars.length-1;i++) {

System.out.println(myCars[i]);

}

System.out.println("\*\*\*\*\*\*");

// To read values from array using for each

for(String myCar : myCars) {

System.out.println(myCar);

}

============================================================= END OF CLASS ========================================================================================