*CONTROL structures (or) Conditional Statements*

1. Define control structures?
2. What are the different types of control structure?
3. Explain about Branching (or) Decision making control structures?
4. Explain about “if” statement?
5. Write the syntax for “if” statement?
6. What is the value when the boolean\_expression is true in “if” statement?
7. Explain about “If – else” statement?
8. Write the syntax for if-else statement?
9. Explain about “If-else-if” statement?
10. Write the syntax for “if – else-if” statement?
11. What are the major rules should follow when we writing if, if-else, else-if statements?
12. Explain about “nested If – else” statement?
13. Write the syntax for “nested if – else-if” statement?
14. Explain about “switch” statement with syntax?
15. Write the syntax for “switch” statement?
16. What is the difference between switch and nested if else statement?
17. What is an optional case for switch statement?
18. Explain about “default” keyword?
19. If none of the case is executed in switch statement then what statement will execute?
20. Write the flow of control for “switch” statement?
21. Is it mandatory that every case in “switch” statement needs a break keyword?
22. If no break keyword is there in case statement then what will happens?
23. What are the types of variables which are used in “switch” statement?
24. What type of values allowed for case statements?
25. What are the major rules for writing “switch” statement?
26. Define a looping structure?
27. Explain about looping structures?
28. What are the different types of looping structures?
29. Define “while loop”?
30. Explain about “while – loop”?
31. Write the syntax for “while – loop” statement?
32. What are the differences between “if” and “while loop” statement?
33. Write the flow of control for “while - loop” statement?
34. Explain about “do...while loop” with syntax?
35. Write the syntax for “do…while – loop” statement?
36. Write the flow of control for “do-while - loop” statement?
37. Define a “for” loop?
38. Explain about “for” loop?
39. Write the syntax for “for – loop” statement?
40. Write the flow of control for “for-loop” statement?
41. Explain about initialization step in for loop statement?
42. Explain about boolean\_expression in for loop statement?
43. Explain about enhanced for loop?
44. Write the syntax for “enhanced for – loop” statement?
45. What is a “declaration” statement is in enhanced for loop statement?
46. What is an expression statement is in enhanced for loop statement?
47. Explain about break and continue statements?
48. What is the use of “break” keyword?
49. What are the major rules for writing “break” keyword?
50. Explain about “break” keyword?
51. Write the Syntax for break keyword?
52. Why we use for break keyword?
53. What is the use of continue keyword?
54. Write the Syntax for break keyword?
55. How continue keyword will be used in a for loop?
56. How continue keyword will be used in a while/do-while loop?

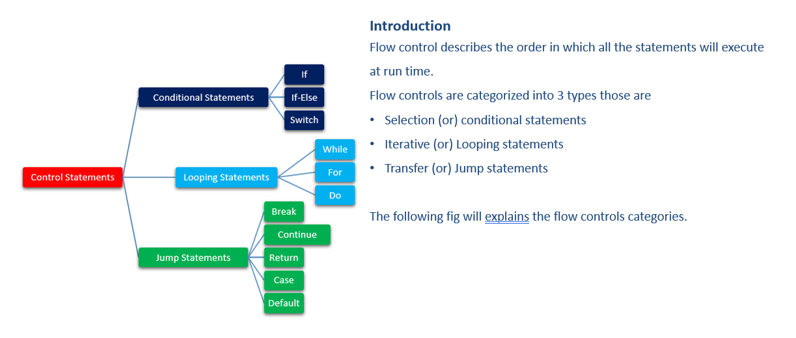
*CONTROL structures (or) Conditional Statements*

1. **Define control structures?**

* Control structures (or) Conditional Statements (or) Flow Controls are used to specify the order of the instructions which are performed in the computer program when the program is executed.

1. **What are the different types of control structure?**

* In java programming language there are two types of control structures they are:
* Branching (or) Decision making structures and
* Looping structures
* Jumping structures



1. **Explain about Branching (or) Decision making control structures?**

* Branching (or) decision making control structures are used to take the decision on the action.
* In java programming language, there are two types of Branching (or) decision making mechanisms They are
* if statements and
* switch statement

1. **Explain about “if” statement?**

* An “if” statement is the simplest statement in the branching statements.
* It consists of “boolean\_expression” followed by one (or) more statements.
* If the “boolean\_expression” is true then the block of statements which inside the “if” statement will be executed otherwise these statements are skipped.

1. **What are the various types of if structures?**

* There are various types of if statement in Java.
  + if statement
  + if-else statement
  + if-else-if ladder
  + nested if statement

1. **Write the syntax for “if” statement?**

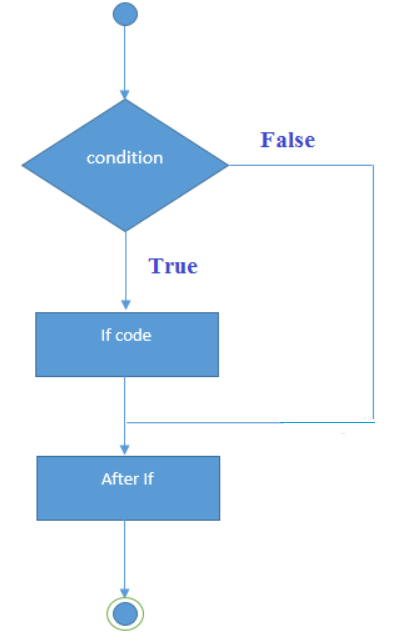
* The Java if statement tests the condition. It executes the if block if condition is true.
* The syntax for “if” statement is shown in below

if (Boolean\_expression)

{

//Statements will execute if the Boolean expression is true

}



1. **What is the value when the boolean\_expression is true in “if” statement?**

* If the boolean\_expression is true in “if” statement then the evaluated values must be non-zero.

1. **Write an example using if statement?**

//Java Program to demonstate the use of if statement.

public class IfExample {

public static void main(String[] args) {

    //defining an 'age' variable

    int age=20;

    //checking the age

    if(age>18){

        System.out.print("Age is greater than 18");

    }  }

}

**Output:**

Age is greater than 18

1. **Explain about “If – else” statement?**

* An “if else” statement contains an optional “else” statement
* It will execute when the “boolean\_expression” is false in if-else statement.

1. **Write the syntax for if-else statement?**

* The Java if-else statement also tests the condition. It executes the if block if condition is true otherwise else block is executed.
* The syntax for if -else statement is shown in below

if (Boolean\_expression)

{

//Executes when the Boolean expression is true

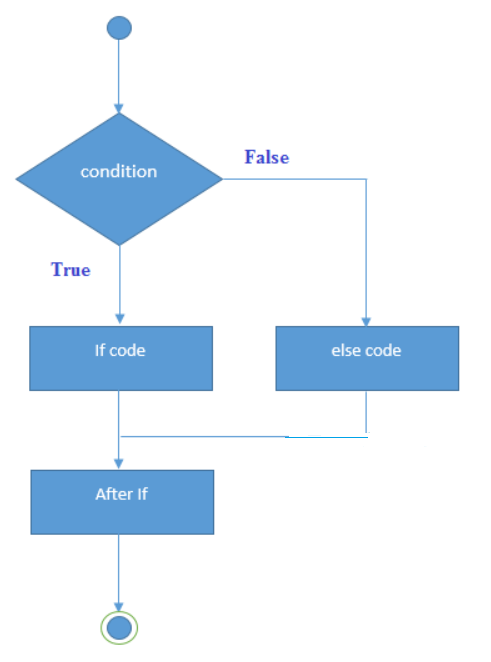
}

else

{

//Executes when the Boolean expression is false

}



1. **Write an examples program using if else statement?**

**Example:1**

//A Java Program to demonstrate the use of if-else statement.

//It is a program of odd and even number.

public class IfElseExample {

public static void main(String[] args) {

//defining a variable

int number=13;

//Check if the number is divisible by 2 or not

if(number%2==0){

System.out.println("even number");

}else{

System.out.println("odd number");

}

}

}

**Output:**

odd number

**Example:2**

A year is leap, if it is divisible by 4 and 400. But, not by 100.

public class LeapYearExample {

public static void main(String[] args) {

    int year=2020;

    if(((year % 4 ==0) && (year % 100 !=0)) || (year % 400==0)){

        System.out.println("LEAP YEAR");

    }

    else{

        System.out.println("COMMON YEAR");

    }

}

}

**Output:**

LEAP YEAR

1. **Explain about “If-else-if” statement?**

* “if....else if” statement is used for testing the various conditions.
* Once an “else if” statement is succeed in “if....else if” statement then remaining “else-if”(or)“else” statements are not tested.

1. **Write the syntax for “if – else-if” statement?**

* The syntax for if -else statement is shown in below

if (Boolean\_expression 1)

{

//Executes when the Boolean expression 1 is true

}

else if (Boolean\_expression 2)

{

//Executes when the Boolean expression 2 is true

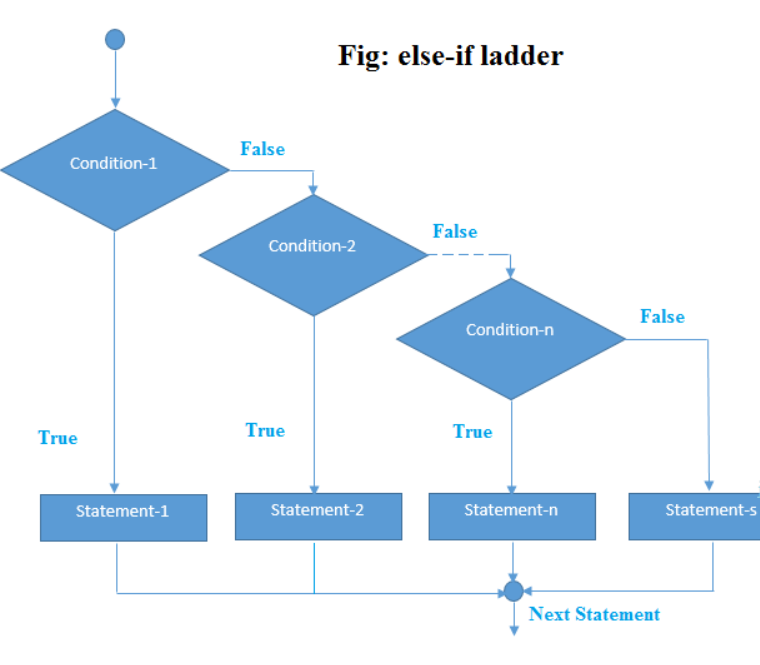
}

else

{

//Executes when none of the above condition is true.

}



1. **What are the major rules should follow when we writing if, if-else, else-if statements?**

* The major rules for writing if, if- else, else-if statements are:
* The “else” statement must be written after the “if” statement and any number of “else if” statements.
* The “else if” statement must be written after the “if” statement and before the “else” statement.

1. **Write an example program using if else if statements?**

**Example:**

//Java Program to demonstrate the use of If else-if ladder.

//It is a program of grading system for fail, D grade, C grade, B grade, A grade and A+.

public class IfElseIfExample {

public static void main(String[] args) {

    int marks=65;

    if(marks<50){

        System.out.println("fail");

    }

    else if(marks>=50 && marks<60){

        System.out.println("D grade");

    }

    else if(marks>=60 && marks<70){

        System.out.println("C grade");

    }

    else if(marks>=70 && marks<80){

        System.out.println("B grade");

    }

    else if(marks>=80 && marks<90){

        System.out.println("A grade");

    }else if(marks>=90 && marks<100){

        System.out.println("A+ grade");

    }else{

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

    }

}

}

**Output:**

C grade

**Example:2**

**Program to check POSITIVE, NEGATIVE or ZERO:**

public class PositiveNegativeExample {

public static void main(String[] args) {

    int number=-13;

    if(number>0){

    System.out.println("POSITIVE");

    }else if(number<0){

    System.out.println("NEGATIVE");

    }else{

    System.out.println("ZERO");

   }

}

}

**Output**:

NEGATIVE

1. **Explain about “nested If – else” statement?**

* The “nested if-else” statement is a mechanism which is used for writing one “if” (or) “else if” statement inside to another “if” (or) “else if” if statement.

1. **Write the syntax for “nested if – else-if” statement?**

* The nested if statement represents the if block within another if block. Here, the inner if block condition executes only when outer if block condition is true.
* The syntax for “nested if-else-if” statement is shown in below

if (Boolean\_expression1)

{

//Executes when the Boolean expression 1 is true

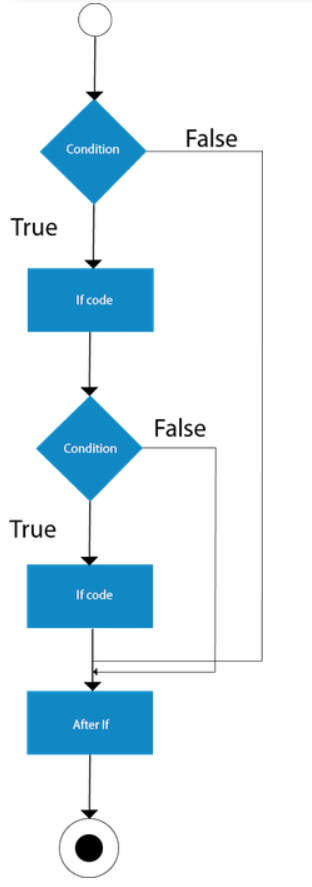
if (Boolean\_expression2)

{

//Executes when the Boolean expression 2 is true

}

}



1. **Write an example using nested if statements?**

**Example:**

//Java Program to demonstrate the use of Nested If Statement.

public class JavaNestedIfExample {

public static void main(String[] args) {

    //Creating two variables for age and weight

    int age=20;

    int weight=80;

    //applying condition on age and weight

    if(age>=18){

        if(weight>50){

            System.out.println("You are eligible to donate blood");

        }

    }

}}

**Output**:

You are eligible to donate blood

**Example 2:**

//Java Program to demonstrate the use of Nested If Statement.

public class JavaNestedIfExample2 {

public static void main(String[] args) {

    //Creating two variables for age and weight

    int age=25;

    int weight=48;

    //applying condition on age and weight

    if(age>=18){

        if(weight>50){

            System.out.println("You are eligible to donate blood");

        } else{

            System.out.println("You are not eligible to donate blood");

        }

    } else{

      System.out.println("Age must be greater than 18");

    }

}  }

**Output**:

You are not eligible to donate blood

1. **Explain about ternary operator?**

* We can also use ternary operator (? :) to perform the task of if...else statement.
* It is a shorthand way to check the condition.
* If the condition is true, the result of ? is returned.
* But, if the condition is false, the result of : is returned.

1. **Write an example using ternary operator?**

public class IfElseTernaryExample {

public static void main(String[] args) {

    int number=13;

    //Using ternary operator

    String output=(number%2==0)?"even number":"odd number";

    System.out.println(output);

}

}

**Output**:

odd number

1. **Explain about “switch” statement with syntax?**

* The Java *switch statement* executes one statement from multiple conditions.
* It is like [if-else-if](https://www.javatpoint.com/java-if-else) ladder statement.
* The switch statement works with byte, short, int, long, enum types, String and some wrapper types like Byte, Short, Int, and Long.
* Since Java 7, you can use [strings](https://www.javatpoint.com/java-string) in the switch statement.
* A “switch” statement is used for to test the variable of equality against a list of values.
* Each value is called a case.
* And the variable being switched on for checking each case.
* It is mostly used for matter of preference which you use.

**Points to Remember**

* There can be *one or N number of case values* for a switch expression.
* The case value must be of switch expression type only. The case value must be *literal or constant*.
* It doesn’t allow [variables](https://www.javatpoint.com/java-variables).
* The case values must be *unique*. In case of duplicate value, it renders compile-time error.
* The Java switch expression must be of *byte, short, int, long (with its Wrapper type),* [enums](https://www.javatpoint.com/java-switch) *and string*.
* Each case statement can have a *break statement* which is optional.
* When control reaches to the [break statement](https://www.javatpoint.com/java-break), it jumps the control after the switch expression.
* If a break statement is not found, it executes the next case.
* The case value can have a *default label* which is optional.

1. **Write the syntax for “switch” statement?**

* The syntax for “switch” statement is shown in below

switch(expression)

{

case value :

//Statements

break; //optional

case value :

//Statements

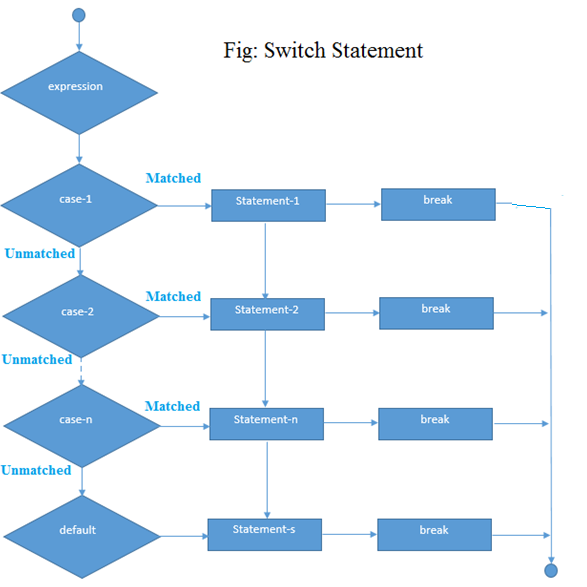
break; //optional

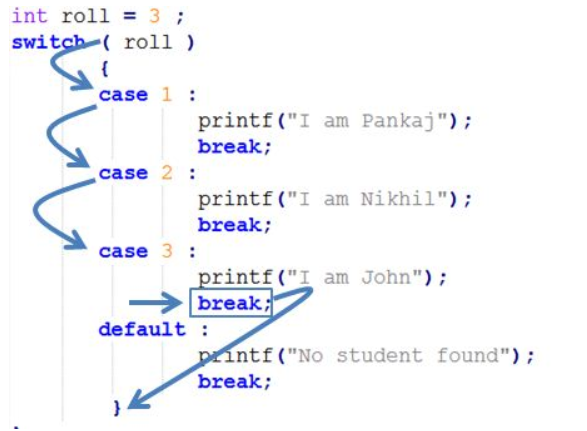
//You can have any number of case statements.

default : //Optional

//Statements

}





1. **Write an example for switch statement?**

**Example: 1**

public class SwitchExample {

public static void main(String[] args) {

    //Declaring a variable for switch expression

    int number=20;

    //Switch expression

    switch(number){

    //Case statements

    case 10: System.out.println("10");

    break;

    case 20: System.out.println("20");

    break;

    case 30: System.out.println("30");

    break;

    //Default case statement

    default:System.out.println("Not in 10, 20 or 30");

    }

}

}

**Output**:

20

**Finding Month Example:2**

//Java Program to demonstrate the example of Switch statement

//where we are printing month name for the given number

public class SwitchMonthExample {

public static void main(String[] args) {

    //Specifying month number

    int month=7;

    String monthString="";

    //Switch statement

    switch(month){

    //case statements within the switch block

    case 1: monthString="1 - January";

    break;

    case 2: monthString="2 - February";

    break;

    case 3: monthString="3 - March";

    break;

    case 4: monthString="4 - April";

    break;

    case 5: monthString="5 - May";

    break;

    case 6: monthString="6 - June";

    break;

    case 7: monthString="7 - July";

    break;

    case 8: monthString="8 - August";

    break;

    case 9: monthString="9 - September";

    break;

    case 10: monthString="10 - October";

    break;

    case 11: monthString="11 - November";

    break;

    case 12: monthString="12 - December";

    break;

    default:System.out.println("Invalid Month!");

    }

    //Printing month of the given number

    System.out.println(monthString);

}

}

**Output**:

7 - July

**Program to check Vowel or Consonant:**

If the character is A, E, I, O, or U, it is vowel otherwise consonant. It is not case-sensitive.

public class SwitchVowelExample {

public static void main(String[] args) {

    char ch='O';

    switch(ch)

    {

        case 'a':

            System.out.println("Vowel");

            break;

        case 'e':

            System.out.println("Vowel");

            break;

        case 'i':

            System.out.println("Vowel");

            break;

        case 'o':

            System.out.println("Vowel");

            break;

        case 'u':

            System.out.println("Vowel");

            break;

        case 'A':

            System.out.println("Vowel");

            break;

        case 'E':

            System.out.println("Vowel");

            break;

        case 'I':

            System.out.println("Vowel");

            break;

        case 'O':

            System.out.println("Vowel");

            break;

        case 'U':

            System.out.println("Vowel");

            break;

        default:

            System.out.println("Consonant");

    }

}

}

**Output**:

20

1. **Explain about Java Switch Statement is fall-through with example?**

* The Java switch statement is fall-through. It means it executes all statements after the first match if a break statement is not present.

**Example:**

//Java Switch Example where we are omitting the

//break statement

public class SwitchExample2 {

public static void main(String[] args) {

    int number=20;

    //switch expression with int value

    switch(number){

    //switch cases without break statements

    case 10: System.out.println("10");

    case 20: System.out.println("20");

    case 30: System.out.println("30");

    default:System.out.println("Not in 10, 20 or 30");

    }

}

}

**Output**:

20

30

Not in 10, 20 or 30

1. **Explain about Java Switch Statement with String and with example?**

* Java allows us to use strings in switch expression since Java SE 7.
* The case statement should be string literal.

**Example:**

//Java Program to demonstrate the use of Java Switch

//statement with String

public class SwitchStringExample {

public static void main(String[] args) {

    //Declaring String variable

    String levelString="Expert";

    int level=0;

    //Using String in Switch expression

    switch(levelString){

    //Using String Literal in Switch case

    case "Beginner": level=1;

    break;

    case "Intermediate": level=2;

    break;

    case "Expert": level=3;

    break;

    default: level=0;

    break;

    }

    System.out.println("Your Level is: "+level);

}

}

**Output**:

Your Level is: 3

1. **Explain about Java Nested Switch Statement with example?**

* We can use switch statement inside other switch statement in Java.
* It is known as nested switch statement.

**Example:**

//Java Program to demonstrate the use of Java Nested Switch

public class NestedSwitchExample {

    public static void main(String args[])

      {

      //C - CSE, E - ECE, M - Mechanical

        char branch = 'C';

        int collegeYear = 4;

        switch( collegeYear )

        {

            case 1:

                System.out.println("English, Maths, Science");

                break;

            case 2:

                switch( branch )

                {

                    case 'C':

                        System.out.println("Operating System, Java, Data Structure");

                        break;

                    case 'E':

                        System.out.println("Micro processors, Logic switching theory");

                        break;

                    case 'M':

                        System.out.println("Drawing, Manufacturing Machines");

                        break;

                }

                break;

            case 3:

                switch( branch )

                {

                    case 'C':

                        System.out.println("Computer Organization, MultiMedia");

                        break;

                    case 'E':

                        System.out.println("Fundamentals of Logic Design, Microelectronics");

                        break;

                    case 'M':

                        System.out.println("Internal Combustion Engines, Mechanical Vibration");

                        break;

                }

                break;

            case 4:

                switch( branch )

                {

                    case 'C':

                        System.out.println("Data Communication and Networks, MultiMedia");

                        break;

                    case 'E':

                        System.out.println("Embedded System, Image Processing");

                        break;

                    case 'M':

                        System.out.println("Production Technology, Thermal Engineering");

                        break;

                }

                break;

        }

    }

}

**Output**:

Data Communication and Networks, MultiMedia

1. **Explain about Java Enum in Switch Statement with example?**

* Java allows us to use enum in switch statement.

**Example:**

//Java Program to demonstrate the use of Enum

//in switch statement

public class JavaSwitchEnumExample {

       public enum Day {  Sun, Mon, Tue, Wed, Thu, Fri, Sat  }

       public static void main(String args[])

       {

         Day[] DayNow = Day.values();

           for (Day Now : DayNow)

           {

                switch (Now)

                {

                    case Sun:

                        System.out.println("Sunday");

                        break;

                    case Mon:

                        System.out.println("Monday");

                        break;

                    case Tue:

                        System.out.println("Tuesday");

                        break;

                    case Wed:

                        System.out.println("Wednesday");

                        break;

                    case Thu:

                        System.out.println("Thursday");

                        break;

                    case Fri:

                        System.out.println("Friday");

                        break;

                    case Sat:

                        System.out.println("Saturday");

                        break;

                }

            }

        }

}

**Output**:

Sunday

Monday

Twesday

Wednesday

Thursday

Friday

Saturday

1. **Explain about Java Wrapper in Switch Statement with example?**

* Java allows us to use four wrapper classes: Byte, Short, Integer and Long in switch statement.

**Example:**

//Java Program to demonstrate the use of Wrapper class

//in switch statement

public class WrapperInSwitchCaseExample {

       public static void main(String args[])

       {

            Integer age = 18;

            switch (age)

            {

                case (16):

                    System.out.println("You are under 18.");

                    break;

                case (18):

                    System.out.println("You are eligible for vote.");

                    break;

                case (65):

                    System.out.println("You are senior citizen.");

                    break;

                default:

                    System.out.println("Please give the valid age.");

                    break;

            }

        }

}

**Output**:

You are eligible for vote.

1. **What is the difference between switch and nested if else statement?**

* Switch statement and nested if else statement must be same but a slightly more efficient and easier to read.

1. **What is an optional case for switch statement?**

* A default case is an optional case for switch statement.
* It must be appear at the end of the switch statement.

1. **Explain about “default” keyword?**

* The “default” keyword is used as an optional case in switch statement.
* The default case is used for performing a task when none of the cases is true in switch statement.
* No “break” keyword is needed in the default case.

1. **If none of the case is executed in switch statement then what statement will execute?**

* If none of the condition is executed in the switch statement then default condition will executes.

1. **Write the flow of control for “switch” statement?**

* When the variable being switched on for checking each case.
* If the value is equal to a case then the block of statements which is followed by that case will execute until a break statement is reached.
* When the break statement is reached in switch statement then the case statement terminates and the flow of control will jumps to the next line following the case statement in switch statement.

1. **Is it mandatory that every case in “switch” statement needs a break keyword?**

* No, not every case in switch statement needs to contain a break keyword.

1. **If no break keyword is there in case statement then what will happens?**

* If no break keyword is appears in the case statement then the flow of control will fall through the subsequent cases until a break keyword is reached.

1. **What are the types of variables which are used in “switch” statement?**

* The types of variables which are used in a switch statement can be only a byte, short, int and char.

1. **What type of values allowed for case statements?**

* The value for a case statement must be the same data type variable which is declared as in boolean\_expression of switch statement and it must be a constant (or) a literal.

1. **What are the major rules for writing “switch” statement?**

* The major rules for writing switch statement are:
* It allows us to write any number of case statements within a switch statement.
* Each case is followed by the value which is compared and a colon.

1. **Define a looping structure?**

* In programming languages, loops are used to execute a set of instructions/functions repeatedly when some conditions become true. There are three types of loops in Java.

1. **Explain about looping structures?**

* There may be a situation when we need to execute a block of code in several numbers of times then we use looping structures.
* It is a mechanism where a task (or) a portion of code is repeated for a certain number of times.
* It is used for decide how many times to take a certain action.

1. **What are the different types of looping structures?**

* In Java programming language, we have three types looping mechanisms they are
* While loop
* Do-while loop
* For loop



1. **Define a “for” loop?**

* A “for” loop is similar to “while” loop but it is just written differently.
* A “for” loop is a repetition control structure.

1. **Explain about “for” loop?**

* The Java for loop is used to iterate a part of the program several times.
* If the number of iteration is fixed, it is recommended to use for loop.
* A for statements are used to process a list or such a range of numbers.
* A for loop is useful when you know how many times a task is to be repeated.

1. **What are the different “for” loop?**

* There are three types of for loops in java.
* Simple For Loop
* [For-each](https://www.javatpoint.com/for-each-loop) or Enhanced For Loop
* Labeled For Loop

1. **Explain about simple “for” loop?**

* A simple for loop is the same as [C](https://www.javatpoint.com/c-programming-language-tutorial)/[C++](https://www.javatpoint.com/cpp-tutorial).
* We can initialize the [variable](https://www.javatpoint.com/java-variables), check condition and increment/decrement value. It consists of four parts:
* **Initialization**: It is the initial condition which is executed once when the loop starts. Here, we can initialize the variable, or we can use an already initialized variable. It is an optional condition.
* **Condition**: It is the second condition which is executed each time to test the condition of the loop. It continues execution until the condition is false. It must return boolean value either true or false. It is an optional condition.
* **Statement**: The statement of the loop is executed each time until the second condition is false.
* Increment/Decrement: It increments or decrements the variable value. It is an optional condition.

1. **Write the syntax for “for – loop” statement ?**

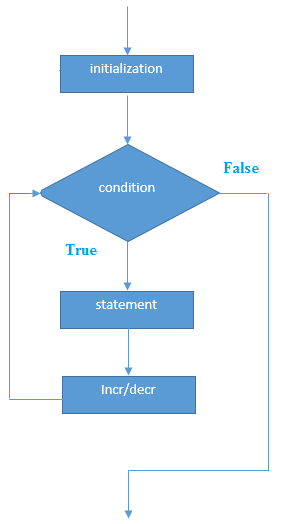
* The syntax for “for – loop” statement is shown in below

for(initialization;condition;incr/decr)

{

//Statements

}



1. **Write the flow of control for “for-loop” statement?**

* Initialization statement will execute first.
* Boolean expression will execute next.
* If the boolean\_expression is true then it will execute a block of statements which is followed by for loop statement.
* After the body executes for loop then the flow of control jumps back to for loop and update the statement.
* The update statement allows you to update any loop control variables.
* After that checks the boolean\_expression.
* If Boolean expression is true then evaluated again.
* This process will repeat until for loop statement is false.

1. **Explain about initialization step in for loop statement?**

* When executing for loop statement, here initialization step is executing first and executes only once.
* This step allows you to declare and initialize any loop control variables.
* You are not required to put a statement here that as long as a semicolon appears.

1. **Explain about boolean\_expression in for loop statement?**

* The Boolean expression is evaluated.
* If it is false then the body of the loop does not execute and flow of control jumps to the next statement.
* If it is true then the loop executes and the process repeats itself (body of loop, then update step, then Boolean expression) until the boolean\_expression is false.
* If the Boolean expression is false then for loop terminates.

1. **Write an example program using for loop?**

//Java Program to demonstrate the example of for loop

//which prints table of 1

public class ForExample {

public static void main(String[] args) {

    //Code of Java for loop

    for(int i=1;i<=10;i++){

        System.out.println(i);

    }  }

}

**Output**:

1

2

3

4

5

6

7

8

9

10

1. **Explain about Java Nested For Loop with example?**

* If we have a for loop inside the another loop, it is known as nested for loop. The inner loop executes completely whenever outer loop executes.

**Example:**

public class NestedForExample {

public static void main(String[] args) {

//loop of i

for(int i=1;i<=3;i++){

//loop of j

for(int j=1;j<=3;j++){

        System.out.println(i+" "+j);

}//end of i

}//end of j

}  }

**Output**:

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

**Pyramid Example 1:**

public class PyramidExample {

public static void main(String[] args) {

for(int i=1;i<=5;i++){

for(int j=1;j<=i;j++){

        System.out.print("\* ");

}

System.out.println();//new line

}

}

}

**Output**:

\*

\*

\* \*

\* \* \*

\* \* \* \*

**Pyramid Example 2:**

public class PyramidExample2 {

public static void main(String[] args) {

int term=6;

for(int i=1;i<=term;i++){

for(int j=term;j>=i;j--){

        System.out.print("\* ");

}

System.out.println();//new line

}

}

}

**Output**:

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

\*

1. **Explain about enhanced for loop?**

* The enhanced for loop was introduced in java 5.
* This is mainly used for Arrays.
* The for-each loop is used to traverse array or collection in java.
* It is easier to use than simple for loop because we don't need to increment value and use subscript notation.
* It works on elements basis not index. It returns element one by one in the defined variable.

1. **Write the syntax for “enhanced for – loop” statement?**

* The syntax for “for – loop” statement is shown in below

for(Type var:array)

{

//Statements

}

1. **What is a “declaration” statement is in enhanced for loop statement?**

* The declaration statement is used for to declare data type of the variable.
* It is used for a block of elements which are compatible with the elements of an array which you are accessing.
* The variable will be available within for block and its value would be the same as the current array element.

1. **What is an expression statement is in enhanced for loop statement?**

* This is used to evaluate the array.
* The expression can be an array variable or method call that returns array elements.

1. **Write a program using enhanced for loop with example?**

//Java For-each loop example which prints the

//elements of the array

public class ForEachExample {

public static void main(String[] args) {

    //Declaring an array

    int arr[]={12,23,44,56,78};

    //Printing array using for-each loop

    for(int i:arr){

        System.out.println(i);

    }

}

}

**Output**:

12

23

44

56

78

1. **Explain about Java Labeled for Loop?**

* We can have a name of each Java for loop.
* To do so, we use label before the for loop. It is useful if we have nested for loop so that we can break/continue specific for loop.
* Usually, break and continue keywords breaks/continues the innermost for loop only.

1. **Write the Syntax for labeled for loop?**

labelname:

for(initialization;condition;incr/decr){

//code to be executed

}

1. **Write an example program using labeled for loop?**

//A Java program to demonstrate the use of labeled for loop

public class LabeledForExample {

public static void main(String[] args) {

    //Using Label for outer and for loop

    aa:

        for(int i=1;i<=3;i++){

            bb:

                for(int j=1;j<=3;j++){

                    if(i==2&&j==2){

                        break aa;

                    }

                    System.out.println(i+" "+j);

                }

        }

}

}

**Output**:

1 1

1 2

1 3

2 1

If you use **break bb;**, it will break inner loop only which is the default behavior of any loop.

public class LabeledForExample2 {

public static void main(String[] args) {

    aa:

        for(int i=1;i<=3;i++){

            bb:

                for(int j=1;j<=3;j++){

                    if(i==2&&j==2){

                        break bb;

                    }

                    System.out.println(i+" "+j);

                }

        }

}

}

**Output**:

1 1

1 2

1 3

2 1

3 1

3 2

3 3

1. **Explain about Java Infinitive For Loop with example?**

* If you use two semicolons ;; in the for loop, it will be infinitive for loop.

**Syntax:**

for(;;){

//code to be executed

}

**Example:**

//Java program to demonstrate the use of infinite for loop

//which prints an statement

public class ForExample {

public static void main(String[] args) {

    //Using no condition in for loop

    for(;;){

        System.out.println("infinitive loop");

    }

}

}

**Output**:

infinitive loop

infinitive loop

infinitive loop

infinitive loop

infinitive loop

ctrl+c

Now, you need to press ctrl+c to exit from the program.

1. **Define “while loop”?**

* The [Java](https://www.javatpoint.com/java-tutorial) while loop is used to iterate a part of the [program](https://www.javatpoint.com/programs-list) several times.
* If the number of iteration is not fixed, it is recommended to use while [loop](https://www.javatpoint.com/java-for-loop).

1. **Explain about “while – loop”?**

* The basic looping structure in java programming language is “while loop”.
* A while loop statement is similar to a repeated “if” statement.
* If the test condition is true then executes the block of statements inside the “while loop”.

1. **Write the syntax for “while – loop” statement?**

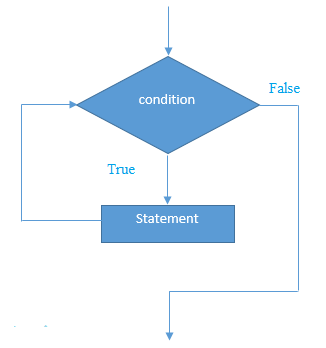
* The syntax for “while – loop” statement is shown in below

while(condition)

{

//Statements

}



1. **Write an example using while statement?**

public class WhileExample {

public static void main(String[] args) {

    int i=1;

    while(i<=10){

        System.out.println(i);

    i++;

    }

}

}

**Output**:

1

2

3

4

5

6

7

8

9

10

1. **Explain about Java Infinitive While Loop with example?**

* If you pass true in the while loop, it will be infinitive while loop.

**Syntax:**

while(true){

//code to be executed

}

**Example:**

public class WhileExample2 {

public static void main(String[] args) {

    while(true){

        System.out.println("infinitive while loop");

    }

}

}

Output:

infinitive while loop

infinitive while loop

infinitive while loop

infinitive while loop

infinitive while loop

ctrl+c

Now, you need to press ctrl+c to exit from the program.

1. **What are the differences between “if” and “while loop” statement?**

|  |  |
| --- | --- |
| **If statement** | **While loop statement** |
| * If statement will executes the block   of statements only one time when   the condition is true. | * While loop statement will executes the   block of statements multiple numbers   of times until the condition is false. |

1. **Write the flow of control for “while - loop” statement?**

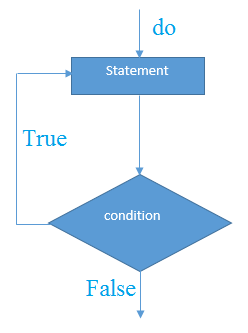
* When executing while loop statement, if the “boolean\_expression” is true, and then executes the following statements which are inside the loop.
* This will continue as long as the “boolean\_expression” is false.
* When the “boolean\_expression” is false then the loop body will be skipped and the first statement after the while loop will be executed.

1. **Explain about “do...while loop” with syntax?**

* A “do ... while loop” is similar to while loop but except that the test condition is checked at the end of the loop.
* The Java do-while loop is used to iterate a part of the program several times.
* If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use do-while loop.
* The Java do-while loop is executed at least once because condition is checked after loop body.

**Syntax:**

1. do{
2. //code to be executed
3. }while(condition);



1. **Write an example using do-while loop?**

public class DoWhileExample {

public static void main(String[] args) {

    int i=1;

    do{

        System.out.println(i);

    i++;

    }while(i<=10);

}

}

**Output**:

1

2

3

4

5

6

7

8

9

10

1. **Write the syntax for “do…while – loop” statement?**

* If you pass true in the do-while loop, it will be infinitive do-while loop.
* The syntax for “do…while – loop” statement is shown in below

**Syntax:**

do{

//code to be executed

}while(true);  );

1. **Write the flow of control for “do-while - loop” statement?**

* When executing do-while loop statement then notice that the Boolean expression appears at the end of the loop, so that the statements in the loop will executes first once before the Boolean expression is tested.
* If the Boolean expression is true, the flow of control jumps back to do statement and executes the following statements which are inside the loop.
* This process repeats until the Boolean expression is false.

1. **Write a program using Java Infinitive do-while Loop?**

public class DoWhileExample2 {

public static void main(String[] args) {

    do{

        System.out.println("infinitive do while loop");

    }while(true);

}

}

**Output**:

infinitive do while loop

infinitive do while loop

infinitive do while loop

ctrl+c

Now, you need to press ctrl+c to exit from the program.

1. **Explain about break and continue statements?**

* In Java programming language provides two commands to control looping statements they are:
* break -- exit form loop or switch.
* continue -- skip 1-iteration of loop.

1. **What is the use of “break” keyword?**

* The “break” keyword is used to stop the entire loop.

1. **What are the major rules for writing “break” keyword?**

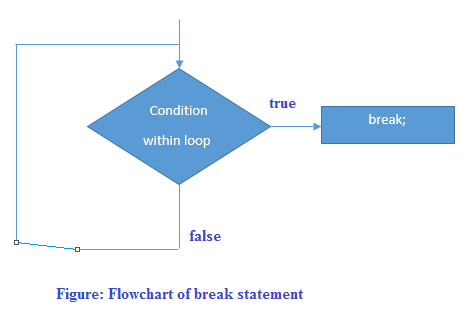
* The break keyword must be used inside any loop or a switch statement.

1. **Explain about “break” keyword?**

* The break keyword will stop the execution of the innermost loop and start executing from the next line of code after the block.
* When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
* The Java break statement is used to break loop or [switch](https://www.javatpoint.com/java-switch) statement. It breaks the current flow of the program at specified condition. In case of inner loop, it breaks only inner loop.
* We can use Java break statement in all types of loops such as [for loop](https://www.javatpoint.com/java-for-loop), [while loop](https://www.javatpoint.com/java-while-loop) and [do-while loop](https://www.javatpoint.com/java-do-while-loop).

**Syntax**:

* + jump-statement;
  + break;



1. **Write a program using Java Break Statement with Loop?**

//Java Program to demonstrate the use of break statement

//inside the for loop.

public class BreakExample {

public static void main(String[] args) {

    //using for loop

    for(int i=1;i<=10;i++){

        if(i==5){

            //breaking the loop

            break;

        }

        System.out.println(i);

    }

}

}

**Output**:

1

2

3

4

1. **Write a program using Java Break Statement with Inner Loop?**

* It breaks inner loop only if you use break statement inside the inner loop.

**Example:**

//Java Program to illustrate the use of break statement

//inside an inner loop

public class BreakExample2 {

public static void main(String[] args) {

            //outer loop

            for(int i=1;i<=3;i++){

                    //inner loop

                    for(int j=1;j<=3;j++){

                        if(i==2&&j==2){

                            //using break statement inside the inner loop

                            break;

                        }

                        System.out.println(i+" "+j);

                    }

            }

}

}

**Output**:

1 1

1 2

1 3

2 1

3 1

3 2

3 3

1. **Write a program using Java Break Statement with Labeled For Loop?**

* We can use break statement with a label.
* This feature is introduced since JDK 1.5. So, we can break any loop in Java now whether it is outer loop or inner.

**Example:**

//Java Program to illustrate the use of continue statement

//with label inside an inner loop to break outer loop

public class BreakExample3 {

public static void main(String[] args) {

            aa:

            for(int i=1;i<=3;i++){

                    bb:

                    for(int j=1;j<=3;j++){

                        if(i==2&&j==2){

                            //using break statement with label

                            break aa;

                        }

                        System.out.println(i+" "+j);

                    }

            }

}

}

**Output**:

1 1

1 2

1 3

2 1

1. **Write a program using Java Break Statement in while loop?**

//Java Program to demonstrate the use of break statement

//inside the while loop.

public class BreakWhileExample {

public static void main(String[] args) {

    //while loop

    int i=1;

    while(i<=10){

        if(i==5){

            //using break statement

            i++;

            break;//it will break the loop

        }

        System.out.println(i);

        i++;

    }

}

}

**Output**:

1

2

3

4

1. **Write a program using Java Break Statement in do-while loop?**

//Java Program to demonstrate the use of break statement

//inside the Java do-while loop.

public class BreakDoWhileExample {

public static void main(String[] args) {

    //declaring variable

    int i=1;

    //do-while loop

    do{

        if(i==5){

           //using break statement

           i++;

           break;//it will break the loop

        }

        System.out.println(i);

        i++;

    }while(i<=10);

}

}

**Output**:

1

2

3

4

1. **Write the Syntax for break keyword?**

* The syntax of a break is a single statement inside any loop:

break;

1. **Why we use for break keyword?**

* If the condition is met in switch case then executes the particular block statements
* And the flow of control will continues on the next case clause also.
* To eliminate the above problem by using break keyword.

1. **What is the use of continue keyword?**

* The continue keyword is used in any of the loop control structures.
* It is used for to immediately jump to the next iteration of the loop control structures.

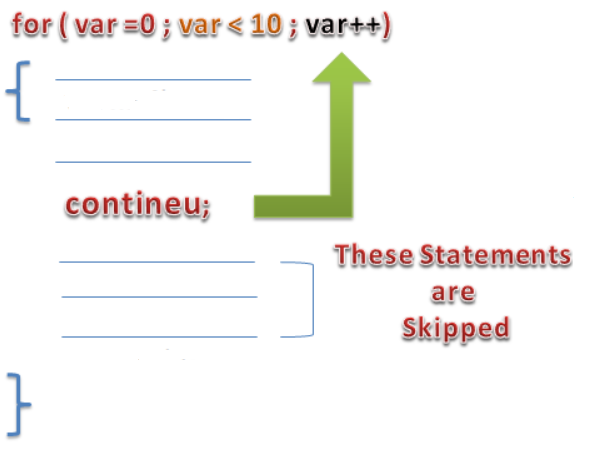
1. **Explain about Java Continue Statement?**

* The continue statement is used in loop control structure when you need to jump to the next iteration of the loop immediately.
* It can be used with for loop or while loop.
* The Java continue statement is used to continue the loop.
* It continues the current flow of the program and skips the remaining code at the specified condition.
* In case of an inner loop, it continues the inner loop only.
* We can use Java continue statement in all types of loops such as for loop, while loop and do-while loop.

**Syntax**:

jump-statement;

continue;



1. **Write a program using Java Continue Statement?**

//Java Program to demonstrate the use of continue statement

//inside the for loop.

public class ContinueExample {

public static void main(String[] args) {

    //for loop

    for(int i=1;i<=10;i++){

        if(i==5){

            //using continue statement

            continue;//it will skip the rest statement

        }

        System.out.println(i);

    }  }

}

**Output**:

1

2

3

4

6

7

8

9

10

* As you can see in the above output, 5 is not printed on the console.
* It is because the loop is continued when it reaches to 5.

1. **Write a program Java Continue Statement with Inner Loop**

* It continues inner loop only if you use the continue statement inside the inner loop.

**Example:**

//Java Program to illustrate the use of continue statement

//inside an inner loop

public class ContinueExample2 {

public static void main(String[] args) {

            //outer loop

            for(int i=1;i<=3;i++){

                    //inner loop

                    for(int j=1;j<=3;j++){

                        if(i==2&&j==2){

                            //using continue statement inside inner loop

                            continue;

                        }

                        System.out.println(i+" "+j);

                    }  }

}  }

**Output**:

1 1

1 2

1 3

2 1

2 3

3 1

3 2

3 3

1. **Write a program using Java Continue Statement with Labeled For Loop?**

* We can use continute statement with a label. This feature is introduced since JDK 1.5.
* So, we can continue any loop in Java now whether it is outer loop or inner.

**Example:**

//Java Program to illustrate the use of continue statement

//with label inside an inner loop to continue outer loop

public class ContinueExample3 {

public static void main(String[] args) {

            aa:

            for(int i=1;i<=3;i++){

                    bb:

                    for(int j=1;j<=3;j++){

                        if(i==2&&j==2){

                            //using continue statement with label

                            continue aa;

                        }

                        System.out.println(i+" "+j);

                    }

            }

}

}

**Output**:

1 1

1 2

1 3

2 1

3 1

3 2

3 3

1. **Write a program using Java Continue Statement in while loop?**

**Example:**

//Java Program to demonstrate the use of continue statement

//inside the while loop.

public class ContinueWhileExample {

public static void main(String[] args) {

    //while loop

    int i=1;

    while(i<=10){

        if(i==5){

            //using continue statement

            i++;

            continue;//it will skip the rest statement

        }

        System.out.println(i);

        i++;

    }

}

}

**Output**:

1

2

3

4

6

7

8

9

10

1. **Write a program using Java Continue Statement in do-while loop?**

**Example:**

//Java Program to demonstrate the use of continue statement

//inside the Java do-while loop.

public class ContinueDoWhileExample {

public static void main(String[] args) {

    //declaring variable

    int i=1;

    //do-while loop

    do{

        if(i==5){

                //using continue statement

                 i++;

            continue;//it will skip the rest statement

        }

        System.out.println(i);

        i++;

    }while(i<=10);

}

}

**Output:**

1

2

3

4

6

7

8

9

10

1. **Write the Syntax for continue keyword?**

* The syntax of a continue is a single statement inside any loop:

continue;

1. **How continue keyword will be used in a for loop?**

* In a for loop, the continue keyword is used to the flow of control will immediately jumps to the update statement.

1. **How continue keyword will be used in a while/do-while loop**?

* In a while loop or do/while loop, the continue keyword is used for the flow of control will immediately jumps to the Boolean expression.

1. **Write a comparison for Java For Loop vs While Loop vs Do While Loop?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Comparison** | **for loop** | **while loop** | **do while loop** |
| Introduction | The Java for loop is a control flow statement that iterates a part of the [programs](https://www.javatpoint.com/java-programs) multiple times. | The Java while loop is a control flow statement that executes a part of the programs repeatedly on the basis of given boolean condition. | The Java do while loop is a control flow statement that executes a part of the programs at least once and the further execution depends upon the given boolean condition. |
| When to use | If the number of iteration is fixed, it is recommended to use for loop. | If the number of iteration is not fixed, it is recommended to use while loop. | If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use the do-while loop. |
| Syntax | for(init;condition;incr/decr){  // code to be executed  } | while(condition){  //code to be executed  } | do{  //code to be executed  }while(condition); |
| Example | //for loop  for(int i=1;i<=10;i++){  System.out.println(i);  } | //while loop  int i=1;  while(i<=10){  System.out.println(i);  i++;  } | //do-while loop  int i=1;  do{  System.out.println(i);  i++;  }while(i<=10); |
| Syntax for infinitive loop | for(;;){  //code to be executed  } | while(true){  //code to be executed  } | do{  //code to be executed  }while(true); |