**Day 5**

**Decision Making Statement – Switch Statement**

* A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case.
* Comparing to if else decision-making statement, switch statement’s readability is more.
* Each case variable should be different.
* default statement will be executed if all other cases didn’t match. It is optional.
* In switch statement, if any of the case meets the condition, then **break statement** is used to terminate from the loop.

**Syntax**

switch(expression)

{

case value1:

Statement;

break;

case value2:

Statement;

break;

default:

Statement;

}

**Example**

// program using Switch statement

class vowels

{

public static void main(String args[])

{

char c = 'm';

switch(c)

{

case 'a':

System.out.println("Character entered is Vowel");

break;

case 'e':

System.out.println("Character entered is Vowel");

break;

case 'i':

System.out.println("Character entered is Vowel");

break;

case 'o':

System.out.println("Character entered is Vowel");

break;

case 'u':

System.out.println("Character entered is Vowel");

break;

default:

System.out.println("Character entered is not a Vowel");

}

}

}

**Output:**

Text

Description automatically generated

**Loop Statements**

Looping statements are used to enable programmers to control the flow of execution by repetitively performing a set of statements if the continuation condition remains true. In Java, there are three types of looping statements available. They are

1. **for loop**
2. **while loop and**
3. **do…** **while loop**

**for loop**

* A for loop is a repetition control structure that allows you to efficiently write a loop that needs to be executed a specific number of times.
* A for loop is useful when you know how many times a task is to be repeated.

**Syntax**

for(initialization; condition; increment/decrement)

{

Statement;

}

**Example**

Class Sample

{

public static void main(String args[])

{

int sum = 0;

for(int i = 0; i <= 10; i++)

{

sum = sum+1;

}

System.outprintln(sum);

}

}

**While loop / Entry Controlled loop**

* While loop in Java comes into use when we need to repeatedly execute a block of statements but doesn’t know the limit.
* Java while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition.
* The while loop can be thought of as a repeating if statement.

**Syntax**

while(condition)

{

Statement;

}

**Example:**

Class Sample

{

public static void main(String args[])

{

int i = 0;

while(i<=10)

{

System.out.println(i); // to print even numbers.

i = i+2;

}

}

}

**do while loop/ Exit Controlled loop**

* The do/while loop is a variant of the while loop.
* This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop if the condition is true.

**Syntax**

do

{

Statement;

}

while(condition);

**Example:**

Class Sample

{

public static void main(String args[])

{

int i = 0;

do

{

System.out.println(i); // to print even numbers.

i = i+2;

}

while(i<=10)

}

}