

~~try, catch, finally~~ → control flow but Java
cannot consider.

Java is

Robust → because of exceptional handling.

Day 21.

M	T	W	T	F	S	S
Page No: 21/02/25					YOUVA	

21/02/25

YOUVA

21/02/25

Leap year → concept

condition: if $\text{num} \cdot 1.4 == 0$ & ($\text{num} \cdot 1.100 == 0$ || $\text{num} \cdot 1.400 == 0$)

1 year → 365.25 days → 1 year 6 hours

2 year → 365.25 days → 1 year 6 hours

3 year → 365.25 days → 1 year 6 hours

4 year → 365.25 days → 1 year 6 hours
4 years 24 hours

so add 24 hours in

$$\therefore 365.2500$$

$$365.2425$$

$$\underline{0.0075}$$

$$0.0075 \rightarrow \text{Error}$$

5th year, then it becomes
leap year which
contains 366 days.

$$900 \times 0.0075$$

$$\therefore 0.0075 \times 100 \text{ years}$$

③

After

→ 100 years - 0.75 days error

3

→ 100 years → $0.0075 \times 100 \text{ years}$

= 1.5 days error.

∴ 3

300 years → 0.0075×300

= 2.25 days.

∴ Each year contains 365 days & 6 hours. so basically
it's not possible to understand anyone that a year
contains 365.25 days so we calculated
0.25 days of each year for 4 years & added in

fifth year. that's why leap years comes. after every
4 years it contains 366 days.

WPP to check whether ^{Year} number is leap year or not

```

→ import java.util.Scanner;
class If3
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the year:");
        int year = sc.nextInt();
        if ((year % 4 == 0 & year % 100 != 0) || (year % 400 == 0))
        {
            System.out.println ("It is leap year");
        }
        else
        {
            System.out.println ("Not a leap year");
        }
    }
}

```

If else if Ladder: →

- It is a decision making statement.
- we use if else if ladder when there are multiple cases conditions
- Eg:- tax slab where if income is less than 12 lakh then you have to pay some tax else if your income is more than 12 lakh then you have to pay some more tax or else if income is less than 12 lakh then you have to pay no tax. don't need to pay any tax.

• Syntax: →

if (condition1)

{

3

else if (condition2)

{

3

else if (condition3)

{

3

else

{

3

- HW: 1 write tax slab program.

- If else if ladder is used when we have to check with multiple conditions.
- An if else if ladder has if block with condition and any no. of condition for else if block and at the end an else block.
- Using else block is not mandatory its optional.
- Working :→
 - The condition present inside if else if ladder must be a boolean type.
 - If the condition in the if block evaluates to true the statements of that block gets executed & control exits the if else if ladder.
 - If the condition of the if block evaluates to false the control transfers to the next condition.
 - This control keeps on happening from one condition block to another until any one condition becomes true.
 - If none of the condition become true the control goes to the else block. If provided. Or it executes nothing.
 - If multiple conditions are true, the condition which evaluates to true first gets executed.

NOTE: → In if else if ladder only one atmost only one block executes at a time.

1,37.00

M	T	W	X	F	S	S	15
Page No.:							
Date:	JSS	GAVA					

Switch Statement :→

- Switch is a decision making statement.

• Syntax:

Switch (value/expression/variable):

{

case value/exp:

{

Statement:

3 [break];

case value/exp:

{

Statement

3 [break];

default :

{

Statement)

3 [break];

}

- A switch statement start with a switch which can accept value or expression or a variable, which has a block which can contain multiple case block ending with a default block.
- A case block can accept value or expression but not a variable *
- Every case block can have a break statement at its end including default block.

Allowed Data Types in Switch →
[byte, short, char, int, String]

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

* Working of Switch: →

- The value passed in the switch is compared with the value present in a case, If it matches. The statement of that case block executes and if the break is used the control transfer out of the switch.
- If the break statement is not used the next other case block executes until it finds the break statement.
- If none of the cases have the break rest all the case block gets executed along with default block.
- If the value of switch does not match the value in the case, the control transfers to the next case and this happens on transferring from one case to another until any case block matches.
- If none of the case block matches the default block gets executed if provided or else it executes nothing.
- Why we cannot pass variable to case?

Ans → If use then error arise like constant expression needed.

Because variable might change in future.

Imp

• Switch allows byte, short, int, char & string.

Anything except this used then "Selector Data type" is not allowed error will occur. error will occur.

• we cannot have 2 case with same value.

NOTE: → In a switch statement we cannot have duplicate case block. (ie more than one case with same value).

- The data types such as byte, short, char, int & string are only allowed in switch statements.
- The data types such as float, long, double, boolean are not allowed.

Break Statement:-

- break is a keyword as well as control transfer statement.
- A break state can be used only inside switch statements & loop statements.
- break ter statement terminate the execution of the switch or loop statements and transfers the control outside.

October is atom → winds season.

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

Eg: class GswitchEx

```
{ public static void main (String args [])
```

```
    byte ip = 1;
```

```
    Switch(ip)
```

```
{
```

```
        case 1:
```

```
{
```

```
            System.out.println ("From case 1");
```

```
            break;
```

```
}
```

```
        case 2:
```

```
{
```

```
            System.out.println ("From case 2");
```

```
            break;
```

```
}
```

```
        case 3:
```

```
{
```

```
            System.out.println ("From case 3");
```

```
            break;
```

```
}
```

```
    default
```

```
{
```

```
    System.out.println ("Default case");
```

```
}
```

```
3
```

M	T	W	T	F	S	S
Page No.:	YOUVA					
Date:						

Eg: → import java.util.Scanner;
 class Switch4
 {

public static void main (String args [])
 {

 System.out.print("Enter A character :");
 char ch = new Scanner(System.in).next().charAt(0);

 if (ch >= 'a' && ch <='z' || ch >= 'A' && ch <='Z')
 {

 switch(ch)
 {

 case 'a':

 case 'e':

 case 'i':

 case 'o':

 case 'u':

 case 'A':

 case 'E':

 case 'I':

 case 'O':

 case 'U':

 {

 System.out.print("It is vowel");
 break;

 }

 default:

 {

 System.out.print("It is consonant");

 }

3

PAGE

- loop →
- ① starting point → initialization
 - ② control the loop → condition
 - ③ to move the → update.
condition false

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

{

3

System.out.println("Not an Alphabet");

3

Loops : → (Loops statements)

- Loop statements helps programmer to execute set of instruction repeatedly.
- There are 4 loops statements in java.

- ① while loop
- ② do while loop
- ③ for loop
- ④ for each loop / advanced for loop / enhanced for loop

Note: Every loop statement has 3 parts

- First one is initialization. It specifies starting point of loop.
- Second one is condition, helps us to control the loop
- Third one is update, it helps us to reach the condition or makes the condition to become false.

while Loop :→

Syntax: initialization
 while (condition)
 {

Statements;

Update;

}

Eg: →

```
int start=0;
while (start < 5)
{
```

System.out.println (" Good Night !");

start++;

}