Pg1

Tuesday, April 28, 2020 12:58 PM

Decision Statements

- A decision statement helps to decide whether to execute a statement or not with the help of a condition.

1. **if** statement
2. **else-if / if-else** statement
3. **else-if ladder** statement
4. **switch** statement

1. if :

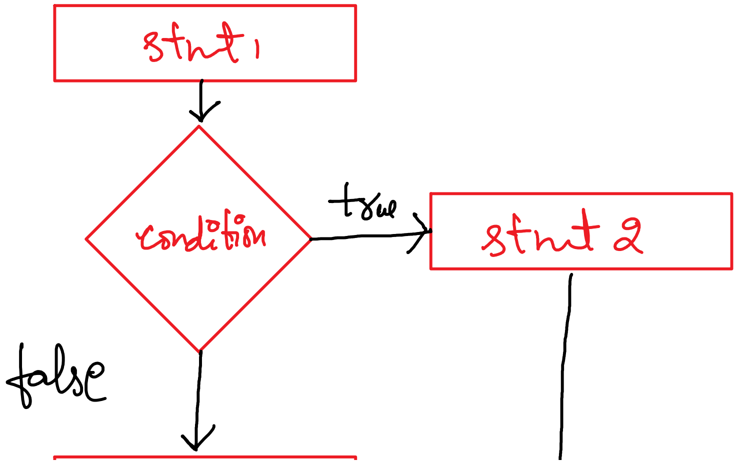
syntax:

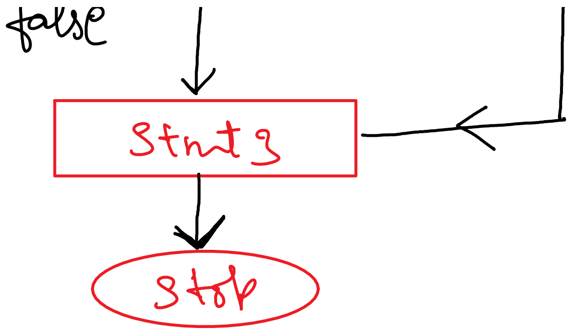
```
1. if (  condition  )
        statement ;

2. if( condition )
   {
            // statements
   }
```

condition → any expression whose result is a boolean type

Flow Diagram

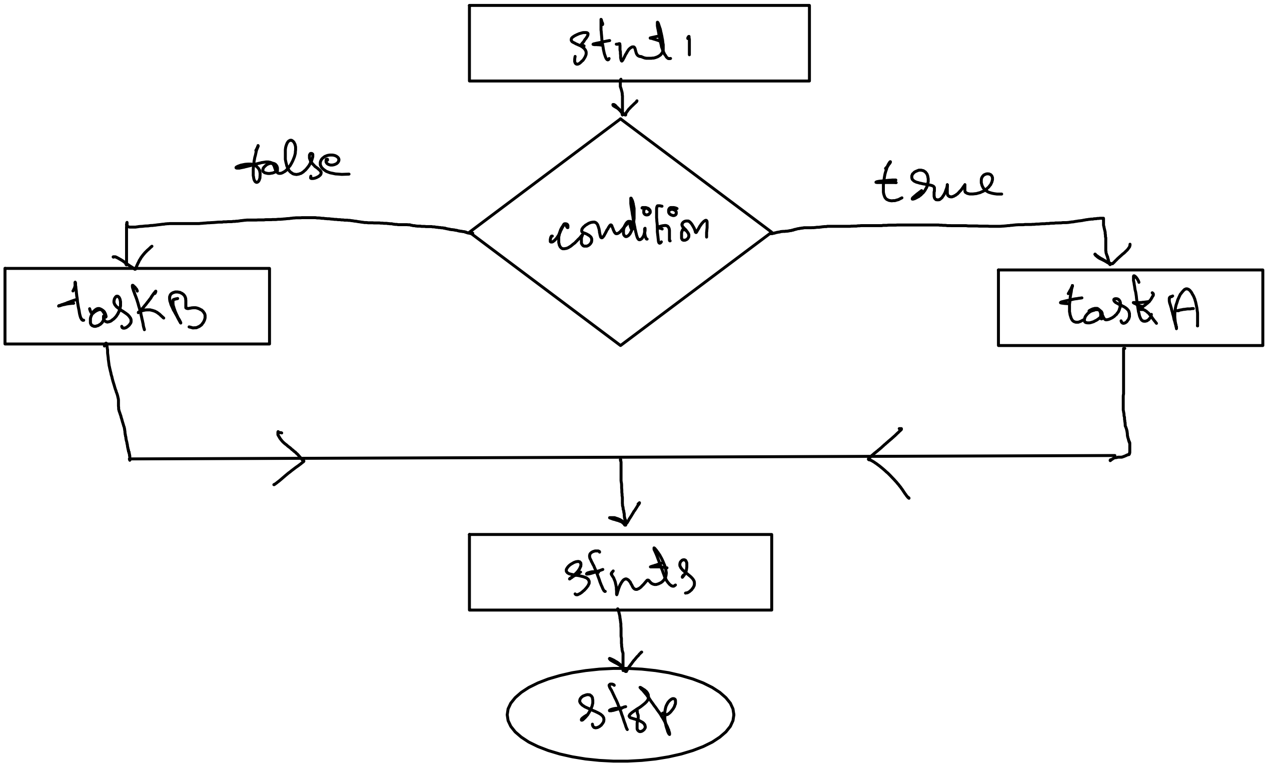



false
Stmt 3
stop

Pg2

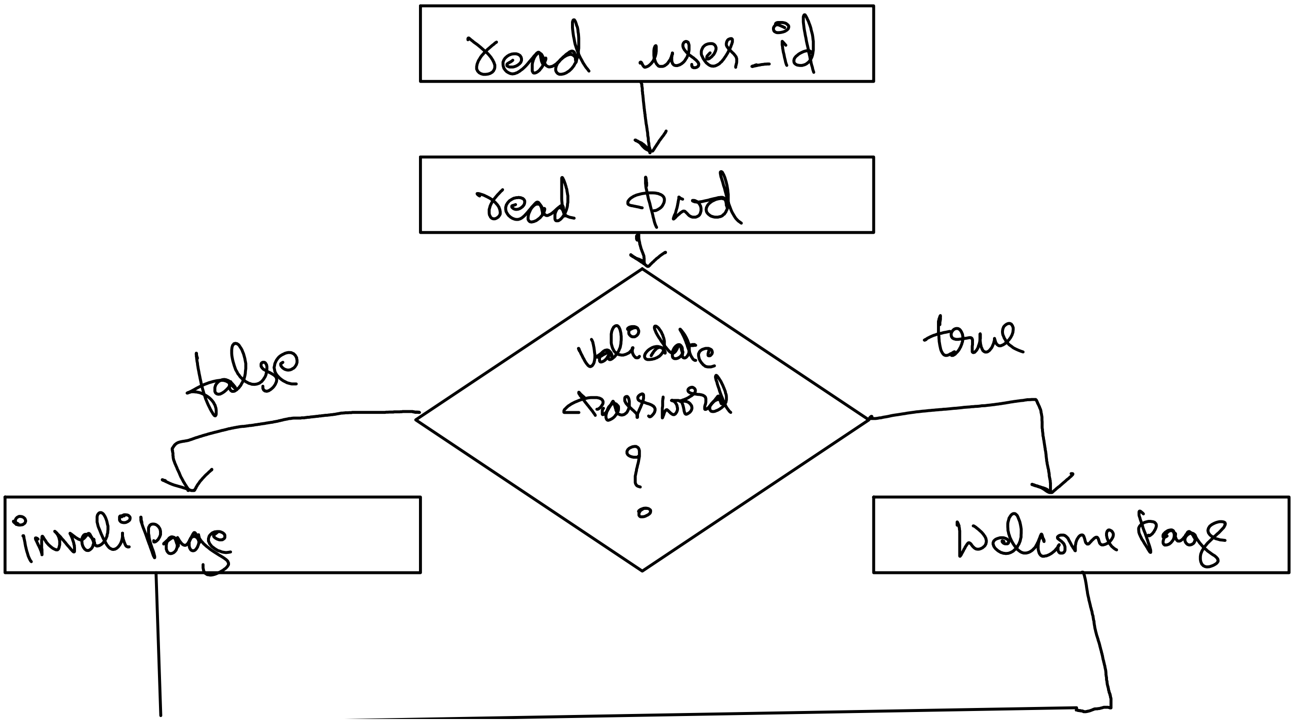
Tuesday, April 28, 2020 1:52 PM

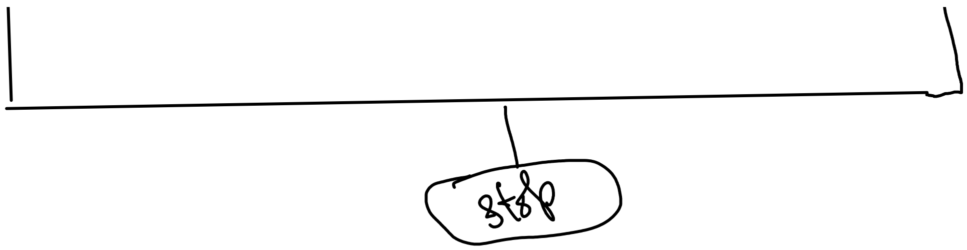
2. <u>if-else statement :</u>



Scenario1 :

Facebook Login





Syntax:

```
1.      if ( condition )
                statement1 ;
        else
                statement2 ;

2.     if ( condition )
        {
            statements ;
        }
        else
        {
            statements ;
        }
```

Task1 :

WAJP to find out the largest of two numbers using decision statement.
(store two numbers and then store the result and print)

Assignment1 :

WAJP to check whether a number is odd or even
(print suitable message)

Assignment2 :

WAJP to check whether a character is a lower or upper case alphabet
(print suitable msg)

Pg3

Wednesday, April 29, 2020 12:37 PM

3. else-if ladder :

   syntax:

```
if ( condition 1 )
{
    statements ;
}
else if( condition 2 )
{
    statements ;
}
else if (condition 3 )
{
    statements ;
}
.
.
.
else
{
    statements
}
```

Note:

1. else block gets executed only if all the if-conditions are false ex refer, **jp/app5/dec_stmt/Z3.java**
2. else block is not mandatory. ex refer, **app5/dec_stmt/Z4.java**

Program1 :

WAJP to obtain largest of 3 numbers using decision statement

Assignments :

1. WAJP to find smallest of 5 numbers

2. WAJP to check whether a character is an :

- ➢ upper case alphabet
- ➢ lower case alphabet
- ➢ a digit
- ➢ special character

3. WAJP to find largest of 5 numbers l

Pg4

Friday, May 1, 2020 12:45 PM

4. switch:

syntax :

```
switch( literal / exp )
{
    case   literal/exp  :    {
                                statements ;
                             }
                             [break ; ]
    case   literal/exp  :    {
                                statements ;
                             }
                             [break ; ]
                  .
                  .
                  .
    default  : {
                    statements ;
               }
}
```

Note :

1. We can pass only byte, short , int, char or String as input to switch. If we pass long, float, double or boolean as input we get CTE.
2. in switch it is not mandatory to use default case.
3. default case will get executed only if all the case is false.
4. **in switch if one case is true, then all the remaining cases will get executed without comparisons(includes default)**.
5. in switch we can compare only using equality operator

break :

- ➢ it is a keyword, it is a control transfer statement.
- ➢ it can be used only inside switch or loop block.
- ➢ If used in switch, it transfers the control outside switch block

Note:

1. it is not mandatory to use break statement
2. we can provide break statement for all the cases, or only to the required cases.

Note:

default case can be written anywhere in the switch.

for examples refer, **jp/app5/dec_stmt/Switch1.java to Switch15.java**

Assignment1 :

1. WAJP to check whether an alphabet is vowel or consonant.

```
switch( literal / exp )
{
    case  literal/exp  :  {
                                  statements ;
                          }
    case  literal/exp  :  {
                                  statements ;
                          }
                .
                .
                .
    default : {
                      statements ;
              }
}
```

-> types : byte, short, int, char, String

== -> equality

true

false

false

```
switch (3)
{
    case 1 :  sop ("idli");
    case 2 :  sop ("dosa");
    case 3 :  sop ("palav");
}
```

3 == 1 - F

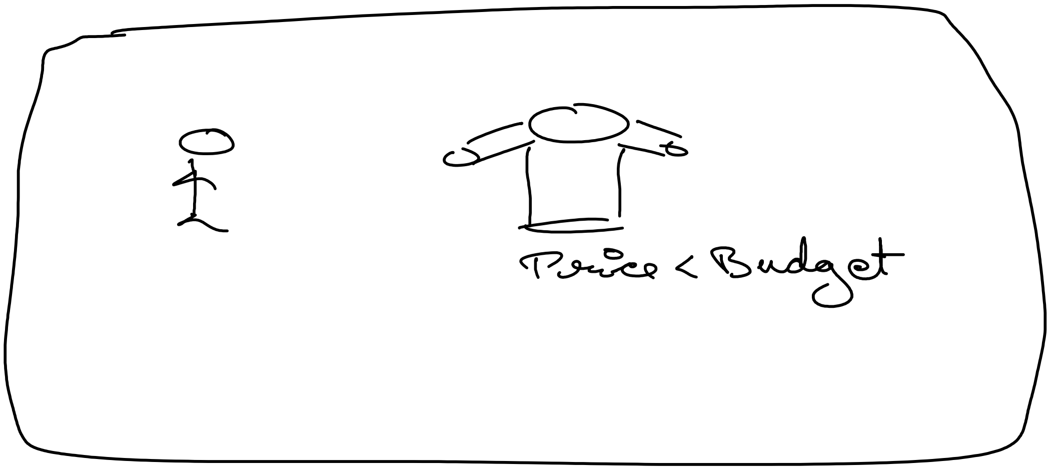
3 == 2 - F

3 == 3 - T

Pg6

Friday, May 1, 2020 2:05 PM

- **Nested Decision statements :**
  - a decision statement written inside another decision statement is known as nested decision statements.
  - we can write any decision statement inside any decision statement. (like, if-else inside if, else-if ladder inside if, switch inside if, etc.)



```
if ( have money & time)
{
        go shopping
        if( price <=' budget)
            {
                buy ;
            }
        else
            {     search ;
            }
}
else
{
```

```
else
  {
        stay home ;
  }
if ( char is alphabet )
  {
        swith ( )
          {
                vowel
                Cons
          }
  }
else {
        char is not alphabet
```