27/4/2024

```
* if ((condition) {      (if body)
          }
  if (condition) {     (if body)
          }
  if (condition) {    (ifbody)
          }
  if (condition)  {    (if body)
         }
  if ((condition)  {   (ifbody)
          }
* if   (condition) {(ifbody); (truebody)
      } else { else body ; (false body)
      }
  if  (condition) { if body; (truebody)
          } else { else body; (false body)
      }
  if  (condition) {(ifbody); (true body)
           } else {elsebody; (false body)
      }
  if (condition) { (ifbody); (true body)
            } else {else body; (false body)
      }
  if (condition)  { ifbody); (true body)
            } else {else body; (false body)
      }
* if  (condition) {ifbody}
   else if (condition) {elsebody}
   else { else body}

  if (condition) {  if body }
    else  if (condition)  {else body}
   else { else body}
```

if ((condition) { if body } else if ((condition) {
if body } else { else body }

if (condition) { if body } else if (condition) {
if body } else { else body }

if (condition) { ifbody } else if (condition) {
ifbody } else { else body }

* instace related to object
instace related to object
instace related to object
instace related to object
instace related to object

* Static related to class
static related to class
Static related to class
static related to class
static related to class

* single Dimansional Array syntax:-

data type array_name[];
data type array_name[];
data type array_name[];
data type array_name[];
data type array_name[];

* multiDimentional Array syntax:-

data type array_name[][];
data type array_name[][];

```
datatype   array_name [][];
datatype   array_name [][];
```

* Constuctor :- Test t = new Test()

1. constuctor name and class name must be same.
2. constructor able to take parameters.
3. constructor not allowed return type.

1. constuctor name and class name must be same.
2. constructor able to take parameters.
3. constructor not allowed return type.

4. constuctor name and class name must be same.
2. constructor able to take parameters
3. constructor not allowed return type

1 constructor name and class name must be same.
2 constructor able to take parameters
3 constructor not allowed return type

1 constructor name and class name must be same
2 constructor able to take parameters
3 constuctor not allowed return type

* Switch (expression)

```
          case lable1:
       sout( );
     break;
          case lable2:
       sout ( );
     break;
          default:
       sout ();
```

```
2  switch (expression)
        case lable1:
            sout();
            break ;
        case lable2:
            sout();
            break ;
        case lable3:
            sout();
            break ;
        default :
            sout();
```

```
(3)  switch (expression)
        case lable1:
            sout();
            break ;
        case lable2:
            sout();
            break ;
        case lable3:
            sout();
            break ;
        default:
            sout();
```

```
3  switch (expression)
        case lable1:
            sout();
            break ;
        case leble2:
            sout();
            break ;
        case lable3:
            sout();     default :   sout();
```

```
4  switch (expression)
        case lable1:
            sout();
            Break ;
        case lable2:
            sout();
            break;
        caselable3:
            sout();
            brack;      default :   sout();
```