Kaushik

Home work - Week - 7

(1) Rules to Declare Constructor;

[1] 1. Constructor name & class name must be same
2. Constructor able to take parameters
3. Constructor not allowed return type.

[2] 1. Constructor name & class name must be same
2. Constructor able to take parameter
3. Constructor not allowed return type.

[3] 1. Constructor name & class name must be same
2. Constructor able to take parameters.
3. Constructor not allowed return type.

[4] 1. Constructor name & class name must be same
2. Constructor able to take parameters.
3. Constructor not allowed return type.

[5] 1. Constructor name & class name must be same
2. Constructor ~~non~~able to take parameters.
3. Constructor not allowed return type.

(2) IF Syntax

[1]
```
IF (condition) {
    // IF body - output IF condition is true
}
```

[2]
```
IF (condition) {
    // IF body - output IF condition is true
}
```

Kaushik

[3] IF (Condition) {
    // IF body - output- IF Condition is true.
}

[4] IF (Condition) {
    // IF body - out-put- IF Condition is true.
}

[5] IF (Condition) {
    // IF body - output IF Condition is true.
}

(3) IF- else Syntax

[1] IF (Condition) {
    // IF body - Output- IF Condition true.
} else {
    // else body IF Condition is not true.
}

[2] IF (Condition) {
    // IF body - out-put IF Condition true.
} else {
    // else body IF Condition is not true.
}

[3] IF (Condition) {
    // IF body - out-put IF Condition true
} else {
    // else body - Out-put IF Condition true
}

[4]
```
if (Condition) {
    // if body - if condition is true out-put
} else {
    // else body if condition is false output.
}
```

[5]
```
if (Condition) {
    // if body - if condition is true out-put
} else {
    // else body if condition is false out-put.
}
```

4 Nested if-else

[1]
```
if (Condition) {
    // output if true
} else if (Condition) {
    // out-put if else if condition true.
} else {
    // out-put if both above condition false.
}
```

[2]
```
if (Condition) {
    // out-put if true
} else if (Condition) {
    // output if first condition not true
} else {
    // out-put if above both condition false
}
```

Kaushik

Pg. 04

[3]
```
if (condition) {
    // output if first condition is true
} else if (condition) {
    // output if first condition false & this is true
} else {
    // if both above are ~~false~~ false.
}
```

[4]
```
if (condition) {
    // output if condition is toure.
} else if (condition) {
    // output if first condition is false and this is true
} else {
    // if both above are false.
}
```

[5]
```
if (condition) {
    // output if condition is true.
} else if (condition) {
    // output if first condition is false and this is true
} else {
    // if both above are false.
}
```

(5)

1. Instance relates to Object
2. Instance relates to Object
3. Instance relates to Object
4. Instance relates to Object
5. Instance relates to Object
6. Instance relates to Object
7. Instance relates to Object
8. Instance relates to Object

(6)

1. Static relates to class
2. Static relates to class
3. static relates to class
4. Static relates to class
5. static relates to class
6. Static relates to class
7. static relates to class
8. static relates to class

(7) Switch statement Syntax

[1]
```
switch (expression) {
    case 1:
        System.out.println();
        breeak;
    case 2:
        System.out.printf();
        breeak;
    case n:
        System.out.println();
        breeak;
    default:
        System.out.println();
}
```

[2]
```
switch (expression) {
    case 1:
        System.out.println();
        breeak;
    Case 2:
        System.out.println();
        breeak;
    Case n:
        System.out.println();
        breeak;
    default:
        System.out.println();
}
```

[3]
```
Switch (expression) {
    Case 1:
        System.out.println();
        breeak;
    Case 2:
        System.out.println();
        breeak;
    Case n:
        System.out.println();
        breeak;
    default:
        System.out.println();
}
```

[4]
```
switch (expression) {
    case 1:
        System.out.println();
        breeak;
    Case 2:
        System.out.println();
        breeak;
    Case n:
        System.out.println();
        breeak;
    default:
        System.out.println();
}
```

[5]
```
switch (expression) {
    Case 1:
        System.out.println();
        breeak;
    case 2:
        System.out.println();
        breeak;
    Case n:
        System.out.println();
        breeak;
    default:
        System.out.println();
}
```

(6)
```
Switch (expression) {
    Case 1:
        System.out.println();
        breeak;
    Case 2:
        System.out.println();
        breeak;
    Case n
        System.out.println();
        breeak;
    default:
        System.out.println();
}
```

8.1 Singl array

[1] int x[] = {int 1, int 2, int 3...};
System.out.println(x[index no.]);

String [] name = {"strg1", "strg2", "strg3"};
System.out.println(name[index no.]);

[2] int x[] = {int 1, int 2, int 3...};
System.out.println(x[index no.]);

String [] name = {"strg1", "strg2", "strg3"};
System.out.println(name[index no.]);
name

[3] int x[] = {int 1, int 2, int 3...};
System.out.println(x[index no.]);

String [] name = {"strg1", "strg2", "strg3"};
System.out.println(name[index no.]);

[4] int x[] = {int 1, int 2, int 3...};
System.out.println(x[index no]);

String [] name = {"strg1", "strg2", "strg3"};
System.out.println(name[index no.]);

[5] int x[] = {int 1, int 2, int 3...};
System.out.println(x[index no]);

String [] name = {"strg1", "strg2", "strg3"...};
System.out.println(name[index no.]);

8.2 Multidimentional Array

Ref: Array 'a'

	0	1
0	100	200
1	300	400
2	500	600

[1] int a[][] = {{100,200}, {300,400}, {500,600}};
System.out.println(a[1][0]); // 200

[2] int a[][] = {{500,200}, {300,400}, {400,500}};
System.out.println(a[2][1]); // 600

[3] int a[][] = new int[3][2];
a[0][0] = 100;
a[0][1] = 200;
a[1][0] = 300;
a[1][1] = 400;
a[2][0] = 500;
a[2][1] = 600;
For(int i=0; i<a.length; i++){
For{int j=0; j<a[i].length; j++){
System.out.println(a[i][j]);
}}

[4] int a[][] = new int[3][2];
a[0][0] = 100;
a[0][1] = 200;
a[1][0] = 300;
a[1][1] = 400;
a[2][0] = 500;
a[2][1] = 600;
System.out.println(a[1][1]); // 400

(5) int a[][] = {{100,200}, {300,400}, {500,600}};
System.out.println(a[2][1]); // 600