

# If-else 3



## Importance of DRY Run:

What is dry Run:

- ① Take a suitable example / test case, and observe the behaviour of code.
- ② Run that case using pen and paper.
- ③ observe the behaviour of code.

\* Indentation is very important: ]

Indentation: formatting code / beautification of code

```
p.s. void main (String [] args)
{
    int n=10;
    System.out.print(n);
}
```

```
p.s. void main (String [] args) {
    — int n=10;
    — S.O.P. (n);
        if( — ) {
            S O P ( );
        }
}
```

## How to solve / Approach a problem:

- ✓ ① Read question carefully and observe what it requires from you.
- ✓ ② Think logic to approach that problem.
- ③ Convert that logic into code.
- ④ Dry Run the code.

### Question: Electricity problem:

Given an Integer (no. of units consume from a customer)  
you have to calculate the amount for that much unit.

Amount is calculated from following rule:

- ① for 1<sup>st</sup> 50 unit  $\rightarrow$  ₹ 0.5 / unit
- ② for next 100 unit  $\rightarrow$  ₹ 0.75 / unit
- ③ for next 100 unit  $\rightarrow$  ₹ 1.20 / unit
- [④ for anything above 250 unit  $\rightarrow$  ₹ 1.50 / unit

unit = 150

$$\begin{array}{l} \underline{A} \\ 150 \rightarrow 50 \} \longrightarrow \text{1<sup>st</sup> 50 unit} \rightarrow \text{Amt} = 0.5 \times 50 = 25 \\ \downarrow \\ 100 \rightarrow \underline{100} \text{ for next 100 unit} \rightarrow \text{Amt} = 0.75 \times \underline{100} = 75 \\ \downarrow \\ \underline{0} \end{array}$$
$$\text{Total} = 25 + 75 = \underline{100}$$

unit = 120

$$\begin{array}{l} A = 120 \\ \downarrow \rightarrow 50. \text{ First 50 unit} \Rightarrow \text{Amt} = 0.5 \times 50 = 25 ₹ \\ \downarrow \\ A = \underline{70} \rightarrow 100, \text{ For next 100, Amt} = 0.75 \times 70 = 52.5 ₹ \\ \downarrow \\ 0 \end{array}$$
$$\text{Total Amount} = 25 + 52.5 = 77.5 ₹.$$

unit = 550

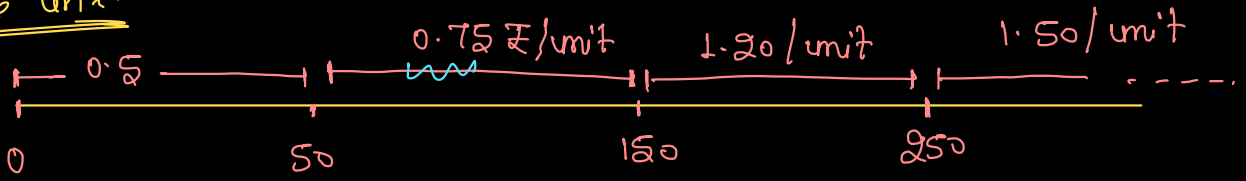
$$\begin{array}{l} A = 550 \rightarrow \textcircled{50} \text{ for first 50 unit} = 0.5 \times 50 = 25 ₹ \\ \downarrow \\ \textcircled{500} \rightarrow \textcircled{100} \text{ for next 100 unit} = 0.75 \times 100 = 75 ₹ \\ \downarrow \\ \textcircled{400} \rightarrow \textcircled{100} \text{ for next 100 unit} = 1.20 \times 100 = 120 ₹ \\ \downarrow \end{array}$$

$$\underline{300} \rightarrow \underline{\text{Above 250}}$$

$$= 1.50 \times 300 = 450 \text{ ₹}$$

$$\text{Total Amt} = 25 + 75 + 120 + 450 = \underline{\underline{670 \text{ ₹}}}$$

Range line



$$\underline{A = 300}$$

$$0 < A \leq 50 \rightarrow \text{Amount} = 0.5 * A$$

$$50 < A \leq 150 \rightarrow \text{Amount} = 0.5 * 50 + 0.75 * (A - 50)$$

$$150 < A \leq 250 \rightarrow \text{Amount} = 0.5 * 50 + 0.75 * 100 + 1.20 * (A - 150)$$

$$\underline{250 < A} \rightarrow \text{Amount} = 0.5 * 50 + 0.75 * 100 + 1.20 * 100 + \underline{1.50 * (A - 250)}$$

$$0.5 * 50 = 25$$

+

$$0.75 * 100 = 75$$

+

$$1.20 * 100 = 120$$

+

$$1.50 * 250 = 75$$

$$\underline{\underline{295}}$$

Scope of a variable  $\rightarrow$

How to know scope of variable -

```

1 // Quiz No. 1
2 import java.util.*;
3
4 public class Main {
5     Run | Debug
6     public static void main(String[] args) {
7         int a = 10;
8         int b = 30;
9
10        if(a > b) {
11            int max = a;
12        } else {
13            max = b;
14        }
15        System.out.println("Max between " + a + " and " + b + " is " + max);
16    }
17 }

```

variable is missing.

Handwritten scope diagram:

```

[
  int a = 10;
  int b = 30;
  [
    int max = a;
  ]
]

```

```

1 // Quiz No. 2
2 import java.util.*;
3
4 public class Main {
5     Run | Debug
6     public static void main(String[] args) {
7         int x = 10;
8         int y = 20;
9         {
10            System.out.print(x + " " + y);
11        }
12        {
13            x = 15;
14            System.out.print(" " + x + " " + y);
15        }
16        System.out.print(" " + x + " " + y);
17    }
18 }

```

How to create a scope / block. →

variable created here have restricted scope

int x = 10 ! 15  
int y = 20 ;

10, 20 - 15, 20 - 15, 20

```

1 // Quiz No. 3
2 import java.util.*;
3
4 public class Main {
5     public static void main(String[] args) {
6         int x = 10;
7         {
8             int y = 20;
9             System.out.print(x + ", " + y);
10        }
11        {
12            x = 15;
13            y = 10;
14            System.out.print(" - " + x + ", " + y);
15        }
16        System.out.print(" - " + x + ", " + y);
17    }
18 }

```

[  
int x = 10  
[  
int y = 20  
] ↑ y is unreachable.  
[

y variable is missing → compilation err

First step is running a code from compiler.

Compiler will check our code syntactically.

→ If syntactically anything is wrong it will throw compile time error.

Example → semicolon is missing.

→ If anything is syntactically correct but conditionally not fit, it will throw runtime error.

Example → dividing from zero.

Error →   
 Syntax Error : Raju sitting chair → grammatically incorrect  
 Semantic Error : Raju is eating chair → grammatically it is correct.

[12/0:]

└ Run time Error.

This is not as expected.

## Compile time Error

If any thing is syntactically incorrect, compiler will catch that error and throw compile time error.

Eg - Missing semicolon  
Missing parenthesis.

We can easily correct it, because compiler will mention correct no. of line

break

Time : 10:50

## Run time Error

After successful compilation, when we run our program if error catch at that time, it is run time error.

Example → divide by 0.

No. of line of error is not correctly mention

```
1 // Quiz No. 4
2 import java.util.*;
3
4 public class Main {
5     Run | Debug
6     public static void main(String[] args) {
7         if(true) {
8             int x = 10;
9             System.out.println("Value of X = " + x);
10            x++;
11        }
12        System.out.println("Value of X = " + x);
13    }
14 }
```

[  
[  
int x = ~~10~~ 11  
]  
]

↳ value of x = 10

↳ x is variable is

if ( TRUE expression ) {

Missing  
→ compile time error

↳ // statements  
}

if expression → TRUE true.  
→ FALSE false.

```
if ( true ) {
```

```
    // statement 1
```

```
} else {
```

```
    // statement 2 ] it this statement is  
~ reachable.
```



```

1 // Quiz No. 5
2 import java.util.*;
3
4 public class Main {
5     public static void main(String[] args) {
6         int a = 0;
7         {
8             b = 10;
9             System.out.println("b = " + b);
10            c = (0) + a;
11            System.out.println("c = " + (b));
12        }
13        a = c + b;
14        System.out.println("a = " + a);
15    }
16 }

```

[

int a = 0;

[

int b = 10;

int c = b + a = c = 10

]

gt is not reachable.

a = c + b

variables are missing.

→ compile time error.

int a = 10;

int b = 20;

int c = 30;

int d = 40;

System.out.println(a + b + " and " + c + d);

## Doubt Session:

constraints are here  $\rightarrow$  Restrictions of thought.

take input  $\rightarrow$  no. of month year.

constraints.

$$\underline{\underline{0 \leq \text{month} \leq 12}}$$

```
if (input < 0) {  
    sys0("invalid input");  
}
```

---