Conditionals & Loops

If statement

* Syntax if-else

```
if (boolean expression T or F){
   // body
} else {
    // do this
 }
```

Ex:

```
int salary = 25400;
if (salary > 10000) {
    salary = salary + 2000;
   } else {
     salary = salary + 1000;
   }
```

* Syntax of Multiple if-else

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

→ if none of the above condition is true then execute this

Loop

* Syntax for loop

```
for( initialisation, condition, incre/decrement){
   // body
}
```

* use for loop when you know how many times the loop is going to run.

* Print no from 1-5

```
for(int num = 1; num <= 5; num += 1){
```

`int num = 1;` → variable num will be initialized to 1

`num <= 5;` → checking if 1 less than or = 5

`num++` , → increamnt till the condition false

* Print no from 1-n

```
Scanner in = new Scanner (System.in);
int n= in.nextInt();
for( int num=1 ; num <=n; num++) {
  System.out.println("Johnson");
}
```

While loop

*Syntax

```
while (condition){
      //body
    }
```

* Initialisation outside while loop
  Condition inside the while loop
  Increcement body of the loop

* Use while loop when you don't know how many times the loop is going to run

Ex1 Print 1 to 5

```
int num= 1;
while(num <= 5){
   System.out.println(num);
  num += 1;
}
```

Do-while loop

* Syntax

```
do {
  } while (condition);
```

* Ex

```
int n=1;
do { System.out.println(n);
```

1
2
3
4

```
do {System.out.println(n);
    n++;
}while(n<=5);
```

3
4
5

When to use do-while loop?
→ In do-while loop is going to execute at least once.

Ex 2

```
int n=1;
do {System.out.println("Hello world");
}while(n!=1);
```

// Hello world

* Here we can see that whatever the while condition it maybe do while loop execute atleast once.

Questions

1. Largest

You will be given 3 numbers

* Step 1 → Input 3 number
Step 2 → Make a as max
Step 3 → check b > max if yes then make b as max
Step 4 → check c > max if yes then make c as max

```
int max=a;
if(b>max){
    max=b;
}
if(c>max){
    max=c;
}
```

* Step 1 → take max=0
Step 2 → chuk a>b if yes then make a max
no then make b max
Step 3 → chuk c > max if yes then make c max.

```
int max=0;
if(a>b){
    max=a;
}else{
```

→ max = Math.max(a,b)

```
    max = a;
} else {
    max = b;
}
if (c > max) {
    max = c;
}
```

→ `max = Math.max(a,b)`

going to return max value b/w a&b

`int max = Math.max(a,b);`

going to return max value b/w (c, a&b)

`int max = Math.max(c, Math.max(a,b));`

2. Input a Characters (letter) and it will tell a character is uppercase & lowercase

Step1 → Take character Input // There is no such thing as in.nextChar

You have to take the string input & first letter of it

```
char ch = in.next().trim().charAt(0);
```

- `in.next()` → Print next word
- `.trim()` → Remove Extra Spaces at the end of the word
- `.charAt(0)` → give me the character at this index of the string
- `char ch =` → Since the result return character we can store it in char type
- Convert the input to char Since when we input it will be string

Step2 : check if input character in between 'a' - 'z'

```
if (ch >= 'a' && ch <= 'z') {
    System.out.println("lowercase");
} else {
    System.out.println("uppercase");
}
```

3. Fibonachi numbers

→ Starts from 0, 1 & by adding the previous 2 number

0, 1, 1, 2, 3, 5, 8, 13 ..

Find n^{th} Fibonachi number for n = 7 answer should be 13

```
Scanner in = new Scanner(System.in);
int n = in.nextInt();
int a = 0;
int b = 1;
int count = 2;
while (count <= n) {
    int temp = b;
    b = b + a;
    a = temp;
    count++;
}
```

4. Big No is given to you and you need to find how many times no 7 is repeating

n = 13 8 5 7 5 7 8 7 9 output 3

```
'i' '3' ;
sum=0;
if(n == 7){
    sum=sum+
}else{
    sum=sumy
```

Step 1: How to get individual digits?

1. Can convert the entire datatype to string datatype then can iterate on it
2. Can use remainder / modulo

```
int n=234;
int count=0;
while (n>0){
    int rem = n%10;
    if (rem==5){
        count++;
    }
    n= n/10;
}
```

5. Given n = 23597 output should be 79532

Remainder approach

Step 1: I will take remainder everytime when no>0 initially ans=0, I will take remainder the answer will be 7

```
23597
```

```
publicclasscalculator{
publicstaticvoidmain(String[]args){
```

```
Scannerin=newScanner(System.in);
//Scannerch=newScanner(System.in);
intans=0;
while(true){
//taketheoperatorasinput
System.out.print("Entertheoperator");
//charop=in.next().trim().charAt(0);

charop=in.next().trim().charAt(0);
if(op=='+'||op=='-'||op=='/'||op=='%'){
//inputtwonumbers
System.out.print("Entertwonumbers:");
intnum1=in.nextInt();
intnum2=in.nextInt();

if(op=='+'){
ans=num1+num2;
}
if(op=='-'){
ans=num1-num2;
}
if(op=='*'){
ans=num1*num2;
}
if(op=='/'){
if(num2!=0){
ans=num1/num2;
}
}
if(op=='%'){
ans=num1%num2;
}
}elseif(op=='x'||op=='X'){
break;
}else{
System.out.println("InvalidOperation!");
}
System.out.println(ans);
}

}
}
```

For - Each loop

* Used to loop through elements in an array.

Syntax

```
for (type variableName : arrayName) {
  // codeblock
}
```

Assignment

* Area of Circle

$A = \pi r^2$ r → radius

* Perimeter of circle

$C = 2\pi r$ r → radius

* Area of triangle

$A = \frac{b \times h}{2}$ b → base, h → height

* Area of rectangle

$A = w \times l$ w → width, l → length

* Perimeter of Rectangle

$P = 2(l + w)$ l → length, w → width

* Area of Isoceles triangle

$A = \frac{b \times h}{2}$

Triangle if 2 sides are equal

* Area of Parallelogram

$A = b \times h$. b = base, h = height

* Parallelogram Perimeter

$P = 2(a + b)$

b → base, a → side

* Area of Rhombus

$\frac{D_1 * D_2}{2}$ D_1 → Diagonal1, D_2 = Diagonal2

*Perimeter of Rhombus

$P = 4a$

a → side

$\frac{D_1 * D_2}{2}$ D_2 = Diagonal?

a → side

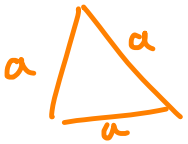
* Area of equilateral triangle

$A = \frac{\sqrt{3}}{4} a^2$ a → side

Perimeter of ET

$P = 3a$

a → side



* Area of square

$A = a^2$

a → side

*Perimeter of square

$P = 4a$

a → side