Taking User Input -

* Scanner Class -

→ Import scanner class in the program

```
import java.util.Scanner;
```

→ Create scanner class object to use its function in the program.

```
Scanner sc = new Scanner(System.in);
```

→ Use scanner's built-in functions.

for eg -

```
int n = sc.nextInt();
float n = sc.nextFloat();
long n = sc.nextLong();
String s = sc.next(); 1 word
```

↳ `sc.nextLine();` more than 1 word.

05 January Saturday 005-360 2019

* Main method -

```
public static void main(int a)
{
    body
}
```

List of parameter

List of parameter holds the no. of input a function/method can take.
Eg- main (int a)
main (int a, float b, char c, double d)
main (boolean b, String s)

* Buffered Reader Class - Same as scanner.

```
import java.io.*;
BufferedReader br = new BufferedReader(
new InputStreamReader(System.in));
int x = ~~br.read(Inte~~
        Integer.parseInt(br.readLine());
float y = Float.parseFloat(br.readline());
```

Operators -

- Arithmetic Operators
- Bitwise Operators
- Assignment Operator
- Comparission Operator
- Logical Operator

* Arithmetic Operators - works on all numbers

+, -, /, *, % → modulus

add subs divide multiply

a = 5, b = 4

c = a+b ⇒ 9
= a-b ⇒ 1
= a/b ⇒ 1
= a*b ⇒ 20
= a%b ⇒ 1

* Bitwise Operators - works on binary number

~ , & , |

not add or

a = 5 ⇒ a & b = 4
b = 3 101 & 110 = 100

⇒ a | b =
101 | 110 = 111

⇒ ~a = 2
~101 = 010

Right Shift - right most value in binary is removed & 0 is added in left
13 >> (1) → place of shifting.

1101 → 0110 → 0011 → 0001 → 0000
13 6 3 1 0

number is reduced by 2; divided by 2.

Left Shift - number is multiplied by 2.
13 << 1 - 26
13 << 2 - 52

* Assignment Operator -
=, +=, -=, *=, /=, %=

For Eg = a = 2 a+=? ⇒ a = a+2

* Comparision Operation = Return type true / false.

==, !=, >, <, <=, >=

* Logical Operator works on expression comparisions.

&&, ||, !

AND OR NOT

Number System

Decimal Number-

124

2 1 0

$10^2 * 1 + 10^1 * 2 + 10^0 * 4$

100 + 20 + 4

124

Decimal Number

1011

3 2 1 0

~~$0 * 2^1 + 1 * 2^1 + 2 * 2^0$~~

$= 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$

$= 8 + 0 + 2 + 1 = 11$

10010110

7 6 5 4 3 2 1 0

$1 * 2^7 + 0 * 2^6 + 0 * 2^5 + 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$

09 January Wednesday
009-356

2019

Conditional Statement -

* if condition -

```
if (condition)
  { statement
  }
```

here, condition output must be of boolean type, i.e 0 for false, 1 for true.

* if-else -

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

here, else block will execute only if the condition given in if block is false.

* if - else if

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

Ternary operator -

condition ? stat1 : stat2

⇒ if (condi.) true that
O/P = stat. 1
else
O/P = stat. 2

JANUARY

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

FEBRUARY

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

MARCH

S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

APRIL

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

MAY

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

JUNE

S	M	T	W	T	F	S
30						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

* Nested if -

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

* Switch statement -

```
switch (choice)
{
    case value1:
        // statement
        break;
    :
    default:
        //statement
}
```

11 January Friday 011-354 2019

Loops -

loops are used to execute a set of statement repeatedly until a particular condition is satisfied.

Types -

- for loop } Entry-control loop
- while / ~~Entry~~ loop }
- do-while loop - Exit control loop.

* for loop -

```
for(initialization; condition; updation)
  {
    statement
  }
```

* while loop -

```
initialization
while (condition)
  {
    //statement
    updation
  }
```

JANUARY

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

FEBRUARY

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

MARCH

S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

APRIL

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

MAY

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

JUNE

S	M	T	W	T	F	S
30						
2	3	4	5	6	7	
9	10	11	12	13	14	
16	17	18	19	20	21	
23	24	25	26	27	28	

* do-while loop-

```
initialization
do
{
  //statement
}
while(condition);
```

⇒ break moves out of the loop and executes the next statement after the loop.

⇒ continue skips the current executing loop and moves to the next loop.

* for each loop-

```
for(datatype var : iterable)
{
}
```