Variable -

used to store value of any data type. 7

int ~~gene~~ age = 10; — Value 8

data type Variable name 9

10

float rate = 2·4821f;

char initials = 'M'; 11

12

Type Casting -

13

Type casting is when you assign a value of one primitive data type to another 14
type.

15

byte → short → char → int → long → float → doubl

16

* Implicit Type casting

* Explicit Type casting. 17

Implicit type casting is done automatically 18
by the java compiler.

JULY						
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			

AUGUST						
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

SEPTEMBER						
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					

OCTOBER						
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		

NOVEMBER						
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

DECEMBER						
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				

Explicit type casting is done forcefully by the programmer. Output is always ceiling value.

Implicit type Casting -

```
    byte x= 4;
    int y = x;
    double z = x;
System.out.println(y,+"\n"+ z);
 Output -  4
           4.0
```

Explicit Type Casting -

```
    double x = 16.043;
      int y = (int) x;
 System.out.println(y);
  Output - 16
```

Arrays

Single Dimensional Array-

- Used to store value of single data item.
- Syn- datatype varName[]= new datatype[size];
- Dynamic memory allocation.

```
int arr[]={1,2,3,4,5,6};
```

arr =

1	2	3	4	5	6

- Different type of allocation-

```
int a[];          1D
int[] a;          1D
int[] a[];        2D
int[] a[][];      3D
int[][] a;        2D
int[][]a[];       3D
```

Multi Dimensinal Array-

2D Array-

- dataType varName[][]=new dataType[size][size2];

-
  ```
  int a[][]={{1,2,3},
             {4,5,6},
             {7,8,9}};
  ```

  a is 3x3 matrix.

Note- new is keyword; used for dynamic (runtime) memory allocation.

-

	0	1	2
0	0,0	0,1	0,2
1	1,0	1,1	1,2
2	2,0	2,1	2,2

a 3x3 (2D) array representation.

String- *Heap Space/String Pool

- Collection of alphanumeric characters enclosed in double quotes (" ").
- Example- "Maneet", "Computer Science", "AI", "123".
- Syntax- String varName = "—".
  ~~Not~~ • Non-Primitive Data Type; index start = 0
  • It is a class.
- If declared using new keyword than at 2 place value is generated string pool(if -nt) & heap area.
- Immutable data Type; if edited in variable than previous value is also present in memory.

Various functions-

- char charAt(int index)
- int length()
- String substring(int begIndex)
- String substring(int begIndex, int endIn)
- boolean contains

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

- boolean equals(Object another)
- boolean is Empty()
- String concate(String str)
- String replace(old, new)
- String[] split(String regex)
- int indexOf(char ch)
- String toLowerCase()
- String toUpperCase()
- String. trim()