

PROGRAMMING PARADISM

6/01/2020

Declaration of variable

a-z
A-Z
0-9
\$
_ (underscore)

→ To start

In java, file name and class name should be same and class name has to start with uppercase letter

Eg:

```
class HelloWorld
{
    public static void main (String [] args)
    {
        System.out.println ("Hello world");
    }
}
```

save the file as HelloWorld.java file extension .java

- No Dynamic Memo Allocation in Java!

all memory stored in Heap.

- Garbage collection algo in JVM (Heap memo)

Datatypes

primitive:- int

float

double

char

boolean

string.

non primitive :-
string
array.

datatype variable ;

Variable naming rules :-

- case sensitive
- begin with letter
- ≠ keyword
- begin with '_', '\$'
- ≠ white space.

No header file.

class is a blueprint and object is reference to the class.

Creating objects

class m:

{

int a;

void func()

{

}

obj

= new

printing output :-

class System.^{o/p string} out.println() - o/p on new line

System.out.print() o/p in single line cursor in same line.

System.out.printf() format (ie no of digits).

e.g.) system.out.println(" kannika");

>>> kannika

1

Q. 2) exp-2.vim ()
{ string x="Computer Science";
s. o.printf ("%s", x)
}
>>> computer

JDK - Java dev kit

JRE (Java Runtime Env)

JDK

- used to run the prog
- used to develop pgm

JVM - task is performed , it reads line by line
Interpreter.

public class Class name

{
 public static void main (String [] args) methods → fun
 {
 }
}

I/O

e.g import java.util.Scanner; ← for user I/p
 or.
 java.util.*;

now create an object for class Scanner

Scanner obj1;

obj1 = new Scanner (System.in); used to accept I/p
 ↑ memory allocation from keyboard

Methods under scanner:

next Int()

next float()

next() → string terminates with white space

nextLine()

next Double()

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

```
public class Prog1
```

```
{
```

```
public static void main (String [] args)
```

```
{
```

```
int x;
```

```
Scanner obj1 = new Scanner (System.in); ← creating object
```

```
System.out.println ("Enter some value");
```

```
x = obj1.nextInt()
```

```
System.out.println ("x=" + x);
```

```
}
```

```
}
```

To find the largest among 3 numbers :-

```
import java.util.Scanner;  
public class Largest  
{  
    public static void main (String [] args)  
    {  
        int a,b,c;  
        Scanner obj1 = new Scanner (System.in);  
        System.out.println ("Enter value");  
        a = obj1.nextInt();  
        b = obj1.nextInt();  
        c = obj1.nextInt();  
        if (a > b && a > c)  
        {  
            System.out.println ("a is big");  
        }  
        else if (b > a && b > c)  
        {  
            System.out.println ("b");  
        }  
        else  
        {  
            System.out.println ("c");  
        }  
    }  
}
```

Type casting

$x = 5.0$ & float convert to int

~~int is higher → lower~~

$$y = (\text{int})x$$

now y is int

public static void main (String [])

}

int a = 10, b = 20, c = 30;

first:

{

second:

{

third:

{
 System.out.println ("a=" + a);

 break second;

 // goto second

}

 System.out.println ("b=" + b);

}

 System.out.println ("c=" + c);

}

» a & c

for ($i=1$; $j=1$; $i < n$; $j < n$; $i++$; $j++$)

Extended for loop.

for (datatype element; array) name

- i) Traverse in forward dir. : standard
- ii) can't access element, as no indexing. : trap
"so avoid using it"
- iii) default inc is 1. : Jumps

e.g.-

int a[] = {5, 6, 7, 8}

for (int element; a)

{
S. O. cout (

D) Write a program to read 5 subject marks and grade the marks using conditional.

```
import java.util.Scanner;  
public class Marks  
{  
    public static void main(String[] args)  
    {  
        int sub1, sub2, sub3, sub4, sub5;  
        // creating an object.  
        Scanner obj1 = new Scanner(System.in);  
        System.out.println("Enter sub1 marks");  
        sub1 = obj1.nextInt();  
        System.out.println("Enter sub2 marks");  
        sub2 = obj1.nextInt();  
        System.out.println("Enter sub3 marks");  
        sub3 = obj1.nextInt();  
        System.out.println("Enter sub4 marks");  
        sub4 = obj1.nextInt();  
        System.out.println("Enter sub5 marks");  
        sub5 = obj1.nextInt();  
  
        System.out.println("Total marks = " + (sub1 + sub2 + sub3 + sub4 + sub5));  
    }  
}
```

2) Write jp to print multiplication table.

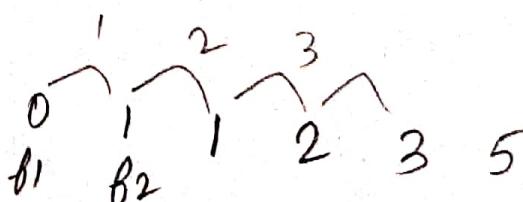
psvm (string [] args)

```
{  
    int n, i;  
    s.o.println("Enter the num of which mult table  
    must be printed ");  
    scanner obj1 = (new scanner (system.in));  
    n = obj1.nextInt();  
    for (i=1; i<=10; i++)  
    {  
        s.p.println(n+"*"+i+" = "+(n*i));  
    }  
}
```

3.

Write jp to find fibb series

```
psvm (  
{  
    int f1=0, f2=1, next
```



4.

To read I/P from user , check if +ve, -ve , 0 , \neq inc user if wants to continue or not.

```
PSVM (String[] args)
```

```
{
```

```
    int n, pos, negt, zero;
```

```
    Scanner obj1 = new Scanner (System.in)
```

```
    char choice;
```

```
do {
```

```
    System.out.println ("Enter any no:");
```

```
    n = obj1.nextInt();
```

```
    if (n > 0)
```

```
        pos++;
```

```
    else if (n < 0)
```

```
        negt++;
```

```
    else
```

```
        zero++;
```

```
    System.out.println ("Enter choice")
```

```
    choice = next();
```

```
    while (choice == 'y' || choice == 'Y')
```

```
        ie if choice is Yes
```

Class and Object

class - blueprint ↗ Attributes
 Methods

object - instance of

- Main should be within a class.

class Demo

```
{  
    int x;  
    doesn't return  
    void setx(y) ← methods.  
    {  
        x = y  
    }  
    returns  
    int getx() ←  
    {  
        return x;  
    }
```

psvm (String [] args)

```
{  
    Demo obj1 = new Demo();  
    obj1.setx(5)  
    z=obj1.getx()  
    S.O.println(z)  
}
```

115

within 1 class

```

class sample1
{
    int x;
    void setx (int y)
    {
        x=y;
    }
    int getx()
    {
        return x;
    }
}

```

Sample 1.java

```

class sample1
{
    public void main (String [] args)
    {
        obj1.setx(5);
        z=obj1.getx();
        System.out.println(z);
    }
}

```

in 2 classes

MODIFIERS

- 1) Public
- 2) Private
- 3) Protected.

CONSTRUCTOR

- whenever we create new object, by default
- To set the default value
- default values
 - int → 0
 - char → '0' (null)

eg. class Demo

```
{  
    private int x;
```

```
    Demo(int y)
```

```
{
```

```
    x = y;  
}
```

]
→ constructor

1) Write JP to find cuberoot by defining attributes and methods in a class.

```
@class Demo
```

```
{  
    int n
```

```
    Demo(int y) {  
        x = y; } → const
```

```
    int cuberoot(int n)
```

```
{  
    return x * x * x → method.  
}
```

```
ps vm (String[] args)
```

```
{  
    Demo obj1 = new Demo(10);
```

```
    s. o. pln (obj1. cuberoot());
```

STATIC VARIABLES:-

- without creating object, we can access.

e.g. Take an eg: 5 students of same clg. Name, Rg.no will be diff but same clg. Instead of allocating memory separately for same clg. make it static.

string name

int rg-no

static ^{clg} string ' clg'

Eg. without static

class Demo

{ int count = 0; normal variable.

Demo()

{ count ++;

s.o. println(count); }

}

p.s.v.m (String[] args)

{ Demo obj1 = new Demo();

Demo obj2 = new Demo();

Demo obj3 = new Demo();

Demo obj4 = new Demo();

}

>> 1

>> 1

>> 1

constructor

with static

class Demo

{

static int count=0;

Demo()

{ count ++

s.o. println(count); }

}

p.s.v.m (String[] args)

{ Demo obj = new Demo();

obj = null;

>> 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

>> 32

>> 33

>> 34

>> 35

>> 36

>> 37

>> 38

>> 39

>> 40

>> 41

>> 42

>> 43

>> 44

>> 45

>> 46

>> 47

>> 48

>> 49

>> 50

>> 51

>> 52

>> 53

>> 54

>> 55

>> 56

>> 57

>> 58

>> 59

>> 60

>> 61

>> 62

>> 63

>> 64

>> 65

>> 66

>> 67

>> 68

>> 69

>> 70

>> 71

>> 72

>> 73

>> 74

>> 75

>> 76

>> 77

>> 78

>> 79

>> 80

>> 81

>> 82

>> 83

>> 84

>> 85

>> 86

>> 87

>> 88

>> 89

>> 90

>> 91

>> 92

>> 93

>> 94

>> 95

>> 96

>> 97

>> 98

>> 99

>> 100

>> 101

>> 102

>> 103

>> 104

>> 105

>> 106

>> 107

>> 108

>> 109

>> 110

>> 111

>> 112

>> 113

>> 114

>> 115

>> 116

>> 117

>> 118

>> 119

>> 120

>> 121

>> 122

>> 123

>> 124

>> 125

>> 126

>> 127

>> 128

>> 129

>> 130

>> 131

>> 132

>> 133

>> 134

>> 135

>> 136

>> 137

>> 138

>> 139

>> 140

>> 141

>> 142

>> 143

>> 144

>> 145

>> 146

>> 147

>> 148

>> 149

>> 150

>> 151

>> 152

>> 153

>> 154

>> 155

>> 156

>> 157

>> 158

>> 159

>> 160

>> 161

>> 162

>> 163

>> 164

>> 165

>> 166

>> 167

>> 168

>> 169

>> 170

>> 171

>> 172

>> 173

>> 174

>> 175

>> 176

>> 177

>> 178

>> 179

>> 180

>> 181

>> 182

>> 183

>> 184

>> 185

>> 186

>> 187

>> 188

>> 189

>> 190

>> 191

>> 192

>> 193

>> 194

>> 195

>> 196

>> 197

>> 198

>> 199

>> 200

>> 201

>> 202

>> 203

>> 204

>> 205

>> 206

>> 207

>> 208

>> 209

>> 210

>> 211

>> 212

>> 213

>> 214

>> 215

>> 216

>> 217

>> 218

>> 219

>> 220

>> 221

>> 222

>> 223

>> 224

>> 225

>> 226

making member static

```
class Demo  
{  
    int count = 0;  
    static void m()  
    {  
        s.o.p("Hello");  
    }  
    psvm()  
    {  
        Demo.  
        Demo.m();  
    }  
}
```

Keyword "This":

Program to take student details:

```
class Demo  
{  
    string name;  
    int rollno;  
    string college = "MSR";  
}
```

} attributes / instances

Demo(string name, int rollno)

```
{  
    name = name; local variables  
    rollno = rollno;  
}
```

Error!, use this.

```

Demo (String name, int rollno)
{
    this.name = name;
    this.rollno = rollno;
}

psvm()
{
    Demo obj1 = new Demo ("xyz", 111);
    Demo obj2 = new Demo ("pqr", 222);
}

```

This. in method , eg:-

```

class Demo
{
    static int count = 0;

    Demo()
    {
        count++;
    }

    int count1()
    {
        this.Demo();
        s.o.println(count);
        this.count1();
    }

    psvm()
    {
        Demo obj1 = new Demo();
        obj1.count1();
    }
}

```

Define class wt reference var length & breadth & initialize ref var through const. Define method to calculate area.

(Class name & const name is same)

class A

{ double l, b;

public A (double len, double bred)

{

l = len; this.len = len;

b = bred; this.bred = bred;

}

} constructor

public void Rect (A obj)

{

int area = l * b;

s.o. pln ("Result area = " + area);

}

class name object name
Passing object to a function

class Rect Demo

{

psvm (string [] args)

{

A obj1 = new A (2, 4);

}

obj1 = Rect();

// fun with 1 para

JP to find sum of 2 no's by passing object to a function

```
class Sum  
{  
    int a, b;  
  
    public int sum (int a, int b)  
    {  
        this.a = a;  
        this.b = b;  
    }  
  
    public void sum (Sum obj1) (int a, int b)  
    {  
        int result1 = obj1.a + obj1.b;  
  
        System.out.println ("result = " + result1);  
    }  
  
}  
  
class AddDemo  
{  
    public void Add ()  
    {  
        int result = a + b;  
        System.out.println ("result = " + result);  
    }  
  
}
```

```
class Add Demo  
{  
    public static void main (String [] args)  
    {  
        Sum obj1 = new Sum (2, 5);  
        obj1.Add (10, 20);  
        obj1.Add ();  
    }  
}
```

• same fun with
same name w/et
diff para —
method overlo
ad

• if same para
it is error!

ARRAYS :- a grp of var of same data type.

java.util.

→ ArrayList class

1) declaring array.

1) datatype var-name[]; int a[];

no size declared

2) datatype[] var-name; int[] a;

2) initialized array: (initially null)

int a[] = {1, 2, 3, 4}.

3) initialization of memory:

int[] a = new int[];

↑ memory allocation

or

int[] a = new int[] {1, 2, 3, 4};

4) length of array

System.out.println(a.length) // 4

5) define size of array

final int x = 5 array size = 5

(to make it constant, through

int[] a = new int[arraysize]. program define)

Q) How to access elements in array.

```
import java.util.*  
class MyArray {  
    public static void main (String [] args) {  
        final int arraysize = 5;  
        int [] a = new int [arraysize];  
        int i; Scanner input = new Scanner (System.in);  
        for (i=0; i < arraysize; i++)  
            { s.o.println ("Enter array value");  
              a[i] = input.nextInt();  
            }  
        s.o.println ("Array is :");  
        for (i=0; i < arraysize; i++)  
            { s.o.println (a[i]);  
            }  
    }  
}
```

Q) To find sum of array elements :

```
import java.util.*  
class MyArray {  
    public static void main (String [] args) {  
        int sum = 0;  
        int i; Scanner input = new Scanner (System.in);  
        for (i=0; i < arraysize; i++)  
            { a[i] = input.nextInt();  
              sum = sum + a[i];  
            }  
        s.o.println ("sum of array elements" + sum);  
    }  
}
```

Passing array to method:

```
psvm() {  
    int a = new int[5] {1, 2, 3, 4, 5};
```

```
    sum(a)
```

```
    A obj1 = new A();
```

```
    obj1.sum(a1);
```

```
    s.o. println(sum)
```

```
}
```

class A {

```
    int sum(int[] a2)
```

```
    { int sum1=0;
```

```
        for (i=0; i<a2.length; i++)
```

```
            sum1 = sum + a2[i];
```

```
        return sum1;
```

```
}
```

JP to find max & min element in an array by passing array to method.

class main()

```
psvm() {
```

```
    int a = new int[5]
```

1, 2, 3, 4, 5

```
; A obj1 = new A();
```

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

```
obj1.read(a); // since passing array
```

```
obj1.max(a); true method
```

all value
as max, run the
loop, update max
value

↳

class A {

```
void read( int[ ] a )
```

```
{ scanner input = new Scanner (System.in); }
```

```
s.o.println("Enter array values");
```

```
for ( i=0; i<a.length; i++ )
```

```
{ a[i] = input.nextInt(); }
```

```
}
```

```
void max( int[ ] a )
```

```
{ int mx = a[0];
```

```
for ( i=0; i<a.length; i++ )
```

```
{ if ( a[i] > mx )
```

```
mx = a[i];
```

```
s.o.println("Max=", mx);
```

```
}
```

```
void min( int[ ] a ) { int mini = a[0];
```

```
for ( i=0; i<a.length; i++ )
```

```
{ if ( a[i] < mini )
```

```
mini = a[i]; }
```

JP to read array of n ele & reverse it.

```
import java.util.*;  
class main {  
    public static void main (String [] args) {
```

```
        int [] a = new int [5];
```

```
        A obj1 = new A();
```

```
        obj1.read (a);
```

```
        obj1.rev_array (a);
```

1	2	3	4	5
---	---	---	---	---

a =

```
class A {
```

```
    void read (int [] a)
```

```
{
```

```
"
```

```
y -
```

```
    void rev (int [] a)
```

```
{
```

```
    int len = a.length;
```

```
    for (i = len - 1; i >= 0; i--)
```

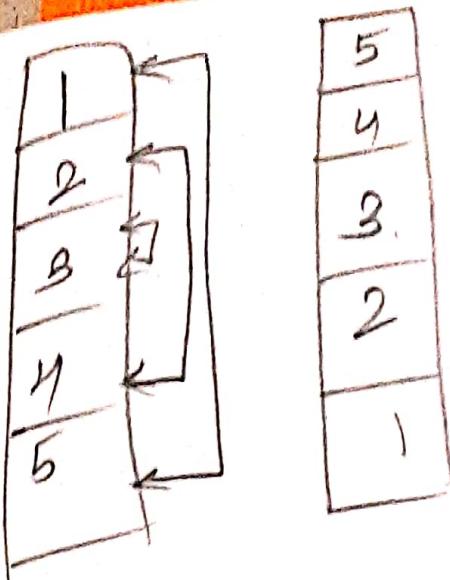
```
{
```

```
    System.out.println (a[i]);
```

```
}
```

```
3
```

1	2	3	4	5
---	---	---	---	---



$\text{temp} = a[i];$
 $a[i] = a[n-i-1];$
 $a[n-i-1] = \text{temp};$

loop has to run $n/2$ times

reversing array by using swapping array:

V.Imp.

void Rev (int [] a)

{ int n = a.length;

for (i=0; i <= (n/2); i++)

temp = a[i];

a[i] = a[n-i-1];

a[n-i-1] = temp;

}

No call by value, reference in java

- ECLIPS ... length can vary , a
can occur only once
• must be last parameter , when we have more parameter

void display (int x, int... array)

• only for 1D Array .

public class Myclass

{ public static void main (String[] args)

{ int a=10, b=20 ; c=30 ; d=40 ;

A object = new A();

int [] a = new int [] { 10, 20, 30, 40, 50 };

obj . display (a)

obj . display (a, b, c, d);

}

Class A

{ void display (int... array)

{ for (i=0; i<array.length ; i++)

{ System.out.println (array [i]) ; }

}

}

}

Program to read an array, n to find sum of elements using variable & arguments, taking input from user.

```
import java.util.*;
public class MyClass
{
    public static void main (String[] args)
    {
        Scanner input = new Scanner (System.in)
        int A[] = new int [A];
        int B[] = new int [B];
        System.out.println ("Enter array element");
        for (i=0; i<size; i++)
        {
            B[i] = input.nextInt();
        }
        A[0].display (B);
    }
}
```

2-D Array

- 1) type [][] array name;
- arrayname = new int [] [];

```
int [] [] array1 = new int [2] [2];
```

2 rows
2 columns

- 2) initializing array element:

```
array1[0] = new int [5]
array1[1] = new int [2]
```

- 3) Accessing elements :

```
for (i=0; i<array1.length; i++)
{
    for (j=0; j<array1[i].length; j++)
    {
        System.out.println (array1[i][j]);
    }
}
```

- 4) Initializing elements :

```
int [] [] array1 = new int [2] [2] { {1, 2, 3, 4, 5}, {6, 7, 8}}
```

```
for (i=0; i<array1.length; i++)
{
    sum = sum + array1[i];
}
System.out.println ("Total sum=" + sum);
```

to find sum of two Matrices :-

```
import java.util.*;  
public class MyClass  
{  
    public static void main (String[] args)  
    {  
        int [][]A = new int [2][2];  
        int [][] B = new int [2][2];  
        Scanner input = new Scanner (System.in);  
  
        System.out.println ("Enter matrix elements:");  
        for (i=0; i<2; i++)  
            for (j=0; j<2; j++)  
                A[i][j] = input.nextInt();  
  
        System.out.println ("Input");  
  
        for mat B:  
  
        for (i=0; i<2; i++)  
            {  
                for (j=0; j<2; j++)  
                    {  
                        ignore and  
                        avoid  
                                          
                    }  
            }  
    }  
}
```

To find Transpose

```
for (i=0; i<2; i++)  
    {  
        for (j=0; j<2; j++)  
            B[j][i] = A[i][j];  
    }
```

```
for (i=0; i<2; i++)  
    {  
        for (j=0; j<2; j++)  
            System.out.println (B[i][j]);  
    }
```

Command line Arguments :-

```
public class CLA;  
{  
    public void psvm (String[] args) {  
        for (int i=0; i<args.length; i++)  
            {  
                System.out.println (args[i]);  
            }  
        System.out.println ("length is :" + args.length);  
    }  
}
```

file name
properties
run
arguments

Arrays Built-in methods :- If A is array

i) $A.sort()$ only ascending order

ii) $A.binarySearch(arrayname, key)$

iii) $equal()$

$array.equal(array1, array2)$

iv) $fill()$

$Array.fill(array name, value)$

ArrayList built-in class

1) Define:

$int[] a = new int[5];$

~~Imp~~ ArrayList <Integers> array = new ArrayList<Integer>();
5 methods under ArrayList set example

a.add(10)
a.add(20)

a.size()

No 'length' \Rightarrow Size()

- a.set[pos, val]
- a.isEmpty()
- a.clear()

import java.util.*;
public class ArrayList1 {

psvm() {
ArrayList<Integer> a = new ArrayList<Integer>();

a.add(100);
a.add(200);

s.o.println(a);

for (int i=0; i<a.size(); i++)
s.o.println(a.get(i));

a.add(30);
s.o.println(a); s.o.println(a.set[0, 100]);

position
value

100, 200

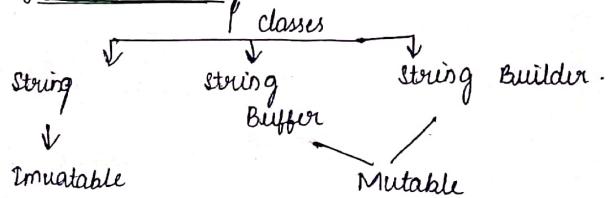
100, 300

Strings - group within double quotes.

char "a" "abc"
char string

package

java.util.String



D) Initialization

a) Using string literals

String s₁ = "Computer"; String s₂ = "Computer"; Same value
only 1 object ie

b) Using new keyword ≈ Arrays

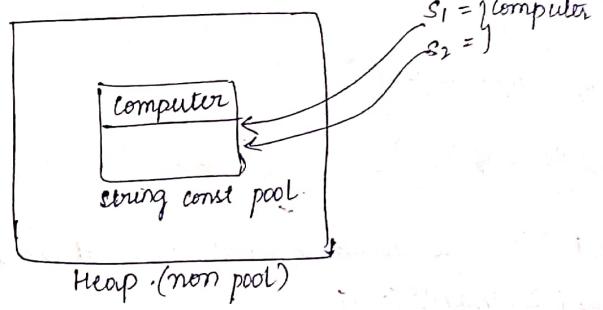
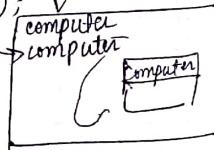
String s₁ = new String("Computer");

All these values(obj) are stored in Heap

String s₂ = new String("Computer");

Two objects.

s₁ refers to heap, not object



∴ String literal → string const pool
new keyword → Heap

public class MyClass

{

```

  public class MyClass
  {
    public static void main(String[] args)
    {
      char[] s1 = new char {'C', 'o', 'm', 'p', 'u', 't', 'e', 'r'};
      String s2 = "Computer";
      String s3 = "MSRUAS";
      String s4 = new String(); // constructor w/ no pa
      String s5 = new String(s3); // MSRUAS
      String s6 = new String(s1); // Computer
      String s7 = new String(s1, 0, 4); // compu
      System.out.println(s4 + s5 + s6 + s7); // ComputerMSRUAS
    }
  }
  
```

array.length

1) string.name.length()

s.o.println(s2.length()) // 8

(computer)

2) charAt() - to access a particular character

s.o.println(s2.charAt(0)); // C

To reverse a string using charAt() :-

```
psvm()
{
    string s2, s1;
```

```
for (int i = s2.length() - 1; i >= 0; i--) {
    s2.charAt(i);
```

```
}
```

convert a string to char array :-

```
char[] a = new char[s];
```

```
s3.getChar(0, 6, a, 0)
```

idea: no starting of array

px for char to array
string invoking

comparing strings:- (on 2 strings)

equals() compare To()

equal ignore case) regionMatches()

= =

1) s.o.println(s2.equals(s3)) // false
(s2.equals(s3)) // true

2) s2.equalsIgnoreCase(s4) // true
(it does not care about upper/lower case).

3) To compare

regionMatches()

if (s3 == s4)

s.o.println("Same");

else

s.o.println("diff");

hello

```
s1. compareTo(s2)
```

MSR

h > M \Rightarrow diff

s2.compareTo(s1)

M < h \Rightarrow diff

$s1.regionMatches(0, s2, 0, 5);$
 starting index
 string invoked.
 starting pos of
 compared string.
 ending pos
 compared string

$s1 = "Computer"$
 $s2 = "COMPUTER"$

Combining two strings:

$s1 = "CSE"$
 $s1.concat(s2) // CSEECE$
 $s2 = "ECE"$

$s1 = "Java Programming"$
 position
 $s1.substring(5) // Programming$
 $s1.substring(0, 6)$
 starting index
 end index

$s1.startsWith("s") // false$
 $s1.startsWith("Ja") // true.$
 $s1.endsWith("ed") // false.$

$s1.indexOf('v') // 2$
 $s1.indexOf('$') // -1$
 $s1.lastIndexof('a') // 10$
 $s1.lastIndexOf("am") // 10$

$x = 10;$
 $String s1 = String.valueOf(x)$
 $System.out.println(s1+s1) // 1010$
 $s.out.println(2+3+"CSE") // 5CSE$
 $s.out.println(2+3+"CSE"+3+5) // 5CSE35$
 $s.out.println(2+3+"CSE"+(3+5)) // 5CSE8$

here x becomes
a string and
not integer.

~~int s1 = Integer.parseInt(String.valueOf(2))~~
~~float s1 = Float.parseFloat(String.valueOf(2.5))~~
~~n = Double.parseDouble(String.valueOf(2.5))~~

String to int, float, double.

$s1.toCharArray()$
 $s1 = "computer"$
 $char[] x = s1.toCharArray()$
 $s.out.println(x); // C$
 $for(i=0; i < x.length; i++)$
 ${ s.out.print(x[i]); }$ // Computer

use extended for loop:

```

for (char ele : x)
{
  s.out.println(ele);
}
  
```

```
s = "Computer";  
s.replace('o', 'O'); // Computer  
s.toUpperCase() // COMPUTER -  
s.toLowerCase() // computer
```

JP to read a string and reverse it , convert
reverse string into char array.

```
import java.util.*;
```

```
public class Java  
{ public static void main()  
{ String str s1;  
int i;  
Scanner input = new Scanner(System.in);  
System.out.println("Enter string");  
String s1 = Input.nextLine();  
for (i = s1.length() - 1; i >= 0; i--)  
{ System.out.println(charAt(i));  
}  
char[] s2 = s1.toCharArray();  
for (char
```

String Builder:

String Buffer java.util

StringBuilder. S1 = new String Builder();

StringBuilder. S2 = new String Builder(10); // 10

"

S3 = "

("Hello") // 16 + 5
= 21

append() ⁸⁴

S3. setLength(44);

insert()

Java Programming

replace()

S1. append("Java")

length()

S2.append("Programming")

delete()

S. println(S1); // Java Programming

setLength()

S1. insert(5, "Hello"); // Java Hello
 programming

charAt()

getChars()

capacity()

s1. delete(5,12); // Java programming

s1. length(); // 16

s1. capacity // 32

s1. setLength(30); // 30

s1. capacity // $(32 \times 2) + 2 =$

s1. getChars(0, 15, array, 0);

for (~~int~~ element: array) → its ASCII value of char.

{
s.o.println(element)

} → java p in char in each block.

s1 = "Java hello program";

String[] s2 = s1.split(" ");

for (String element: s2)

{
s.o.println(element);

// Java

hello

program

s1 = "Hello";

char c = charAt(2) // l c='l'

class character

character.isDigit(c) // False

character.javaIdentifierStart(c) // true

variable
≈
identifier