1.

**a.** System.out.println('j' + 'a' + 'v' + 'a'); //print ASCII code addition of respective character.

**b.** If you try to concatenate any different types of data like integer, character, float with string value, **the result will be a string**

  System.out.print("A" + "B" + 'A' + 10);// ABA10

System.***out***.print("A" + "B" + 'A' + 10.123+10+10l);// ABA10.1231010

**c**. Similar data types are added and then converted to string. 20 and 1.34f will be added and then 21.34 will be concatenated with “A” and “B”,

  System.out.print(20 + 1.34f + "A" + "B");//21.34AB

**d.** if you try to print str.toString() it will not converted to string because str is an object of character array that will **print an address in string format.**

char[] str = { 'a', 'k', 'a', 's', 'h'};

        System.out.println(str.toString());

2.

int $\_ = 5; //legal,var canr be start with $ or \_(underscore)

3. We can use all predefined Java class name and interface name as [identifiers](https://www.geeksforgeeks.org/java-identifiers/).

**int** Runnable = 97;

**int** util=23;//pred. class

**char** Connection='a';//interface

System.***out***.print(Runnable+" "+util+" "+Connection);

4. We can’t use [reserved words as identifiers.](https://www.geeksforgeeks.org/java-identifiers/)

int if = 65; //error

      int else = 97; //error

      System.out.println(if + " : " + else);

5. we can use ‘\_'(under Score) Symbol between digits of numeric literals. Not starting

**double** d1 = 123.456;

**double** d2 = 12\_3.4\_5\_6;

**double** d3 = 12\_3.4\_56;

**float** d4 = 12\_3.4\_56f;

System.***out***.println(d1);

System.***out***.println(d2);

System.***out***.println(d3);

System.***out***.println(d4);

double d1 = \_123 .456;//error

Output:

123.456

123.456

123.456

123.456

6.

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

       System.out.println(arr);// printing the base address of the array

char[] str = { 'a', 'k', 'a', 's', 'h'};

         System.out.println(str.toString());//will print an address in string format.

**char**[] str = { 'a', 'k', 'a', 's', 'h'};

String str2=Arrays.*toString*(str);

System.***out***.println(str2);// [a, k, a, s, h]

OR

System.***out***.println(Arrays.*toString*(str)); [a, k, a, s, h]

        int arr2[5] = { 1, 2, 3, 4, 5 }; //Error bz size and value together

7.

a.

**int** arr1[], arr2[]; //this is legal

**int**[] arr3, []arr4;//Not allowed gives C.T Error

b.

**int** arr1[] = **new** **int**[-1]; //Compile but R.T error

System.***out***.print(arr1.length);

c.

we can specify the array size with char, sort, int, byte but we can not with long, double, string and float size.

**char** s = 65; we can use byte,int

**int** arr1[] = **new** **int**[s];

8.

a. {

for (;;)

            System.out.println("GEEKS"); //print infinetly

     }

b.

for (int i = 0; i < 3;) // need semicolon

            System.out.println("GEEKS");// print infinetly

c.

**for** (**int** i=0;; ) {

System.***out***.println("HIII"); //print infinetly

}

d.

**for** (;2<5; ) {

System.***out***.println("HIII"); //print infinetly

}

e.

**for** (;;i++ ) {

System.***out***.println("HIII"); //print infinetly

}

*In above program if we write any statement after for loop then it will show ‘unrechable code’ C.T error.*

**9.** We can write only static or final or both inside interface

**interface** MyInterface

{

static int *x*=20; **OR** final int x=20; **OR** public static final int x=20;

}

10. We can write:

Public static void main **(String[]args)//without any space**

public static void main(**String... args**)//**three dot allowed only not not less than not greter than**

**11.**

**int i=10;**

**while(i++<=10)**

**{**

**i++;**

**}**

**System.out.println(i);//print 13**

**12.**

**int i=4;**

**SOP(i++);//print 4**

**13: we can compare value of Integer and int**

**Integer i=20;**

**Int j=20;**

**i.equals(j)//true**

**14. Byte is cyclic in java and range is -128 to 127**

**byte f=127;**

**f++;**

**f++;**

**System.out.println(f);//print -127,Byte is cyclic in java**

**15. Without main() we can execute program/code because of Static block.**

**16. list iterater :iterator = arrlist.listIterator(2);start from index 2**

**17. We can override the overloaded method.**

**18. The local variables are not initialized to any default value.**

**15. Octal Number System**

**Any number that start with zero(0) then it will octal**

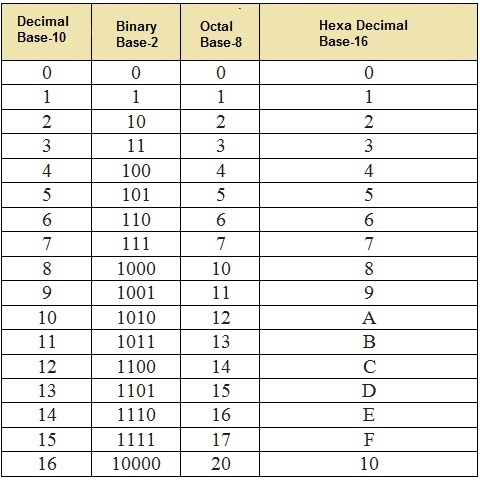
**int a=010;**

**int g=07;**

**System.out.println(a+" "+g);//print 8(octal if start with 0) and 7**

**011 is otcal of decimal 9**

**012 is octal of decimal 10**



**float f=12.5;//Error**

**float f=(float)12.5;**

**float f=12.5f;**

**java not support inner method but class can be.**

**return type neccessary while declare and defined method.**

**In Java Class there are two area:**

**Static area and instance area.**