



Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

balarfcs@iitr.ac.in

<https://sites.google.com/site/balaiiitr/>



In C++,

Example 6

```
int a[3][ ]={1,2,3,4,5,6,7};  
for (int i=0; i<3; i++)  
{ for (int j=0; j<3; j++)  
    cout<<a[ i ][ j ]<<" ";  
    cout<<endl;  
}
```

Output:
Error



Address of $[i1][i2]^{\text{th}}$ location element in a 2D-Array

Given $a[r1][r2]$ array,

$r1$ -No. of rows, $r2$ -No. of Cols.

$0 \leq i1 < r1$ and $0 \leq i2 < r2$, finding the address of $a[i1][i2]$

$\&a[i1][i2] = \&a[0][0] + (i1 * r2 + i2) * e_size;$

In C++,

Example 4

```
int a[ ][3]={1,2,3,4,5,6};  
for (int i=0; i<3; i++)  
{ for (int j=0; j<7; j++)  
  cout<<a[ i ][ j ]<<" ";  
  cout<<endl;  
}
```

Output:

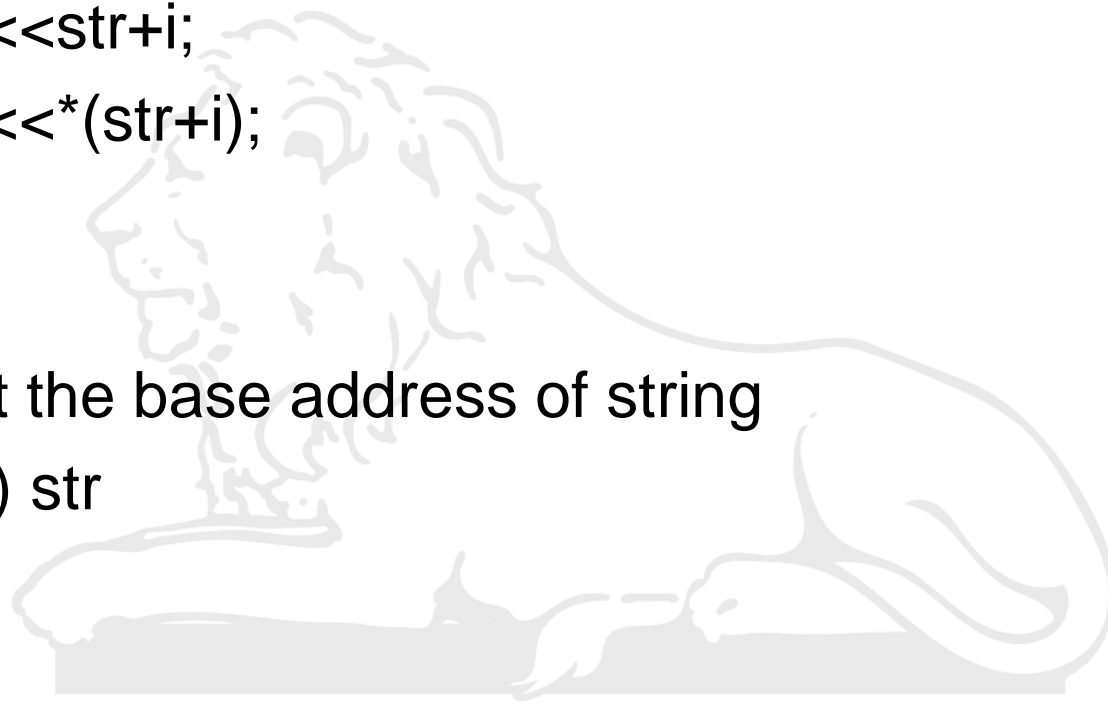
1	2	3	G1	G2	G3	G4
4	5	6	G5	G6	G7	G8
G9	G10	G11	G12	G13	G14	G15

Character String (C++)

```
char *str="IITRoorkee";  
for (int i=0; i<10; i++)  
{   cout<<str+i;  
    cout<<*(str+i);  
}
```

How to print the base address of string

```
(int *) str  
&str
```



Strings in Java

- `Char charArray[] = new char[4];`
`charArray[0]='J';`
`charArray[1]='A';`
`charArray[2]='V';`
`charArray[3]='A';`
- `String str;`
`str=new string("IITRoorkee");`



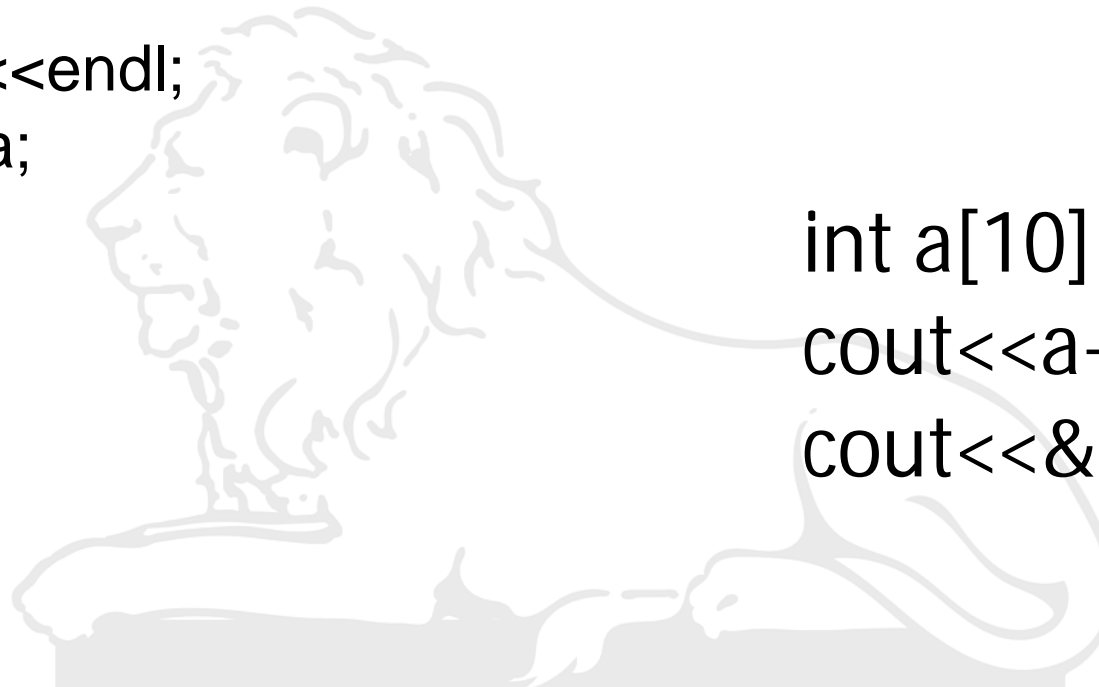
- Write a Java or C++ program to find whether given string is Palindrome or not.



What is the output of following strange code? Why?



```
int a[10];  
cout<<a<<endl;  
cout<<&a;
```



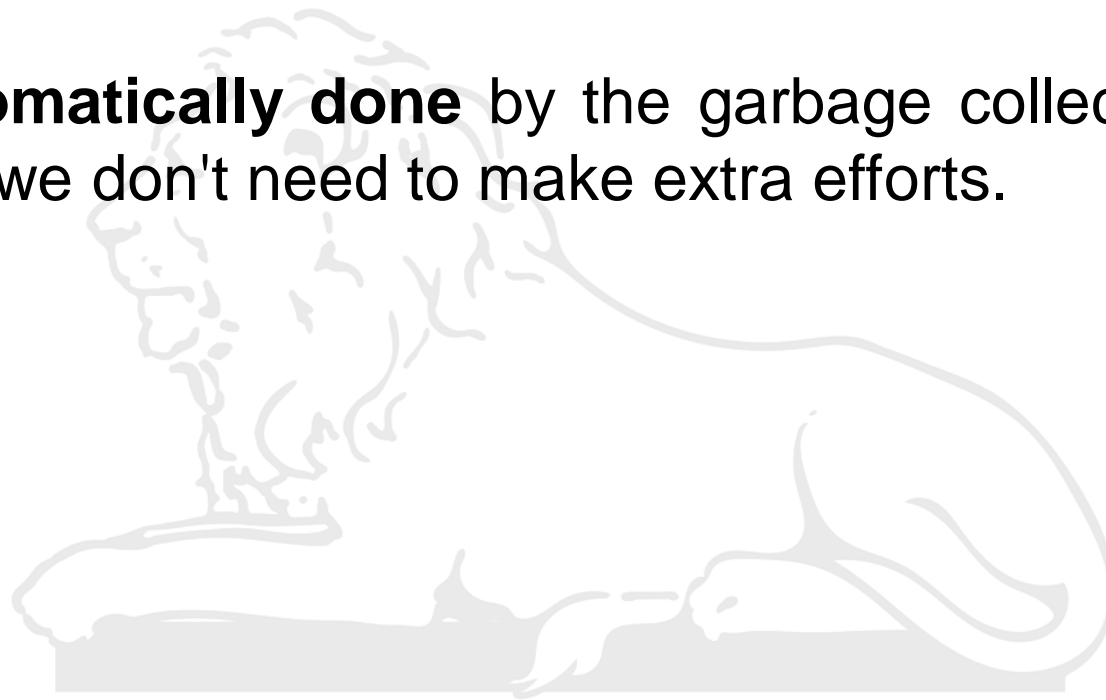
```
int a[10];  
cout<<a+0<<endl;  
cout<<&(a+0);
```


Garbage collection in Java

- In java, garbage means unreferenced objects.
- Garbage Collection is process of reclaiming the runtime unused memory automatically. In other words, it is a way to destroy the unused objects.
- To do so, we were using **free()** function in C language and **delete()** in C++. However, in java it is performed automatically. So, java provides better memory management.

Advantage of Garbage Collection

- It makes java **memory efficient** because garbage collector removes the unreferenced objects from heap memory.
- It is **automatically done** by the garbage collector(a part of JVM) so we don't need to make extra efforts.



How can an object be unreferenced?

- By making the reference Null
- By assigning a reference to another
- By anonymous object etc.



By nulling a reference:

```
Employee e=new Employee();  
e=null;
```



By assigning a reference to another:

```
Employee e1=new Employee();  
Employee e2=new Employee();  
e1=e2; //now the first object referred by e1 is  
        //available for garbage collection
```

