Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

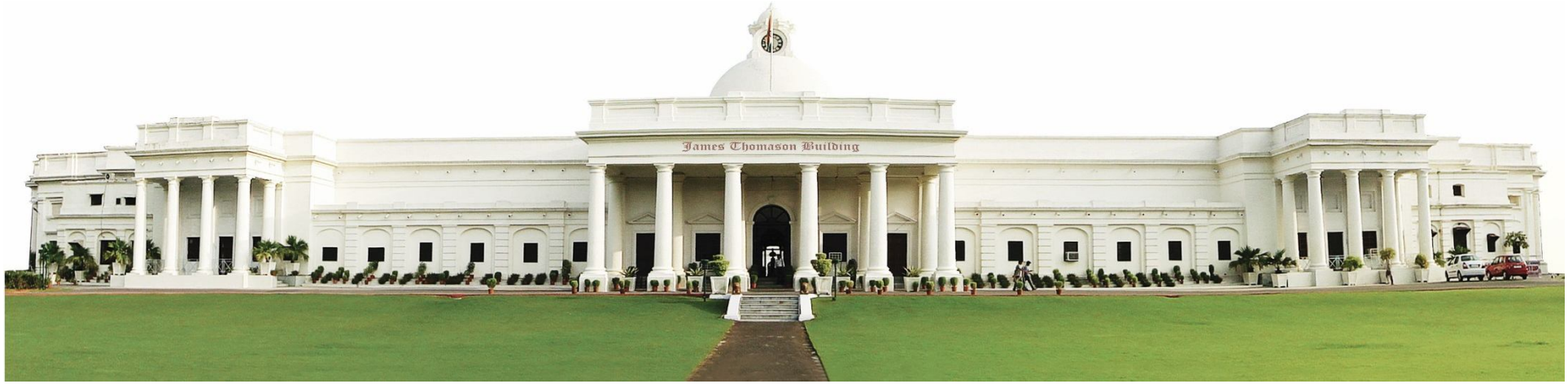
Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

Roorkee 247 667

balarfcs@iitr.ac.in

https://sites.google.com/site/balaiitr/



Find the output of the following C++ Program

```
void change(int* b)
void main()
 { int a[5]={20,30,40,50,60};
    change(a);
    for (int i=4; i>=0; i--)
   cout<<a[i];
 }
void change(int* b)
{ for (int i=0; i<4; i++)
    { *b=*b+1;
       b++;
     }
}
```

```
void main()
 { int a[5]={20,30,40,50,60};
      for (int i=0; i<4; i++)
    { *a=*a+1;
    a++;
     }
for (int i=4; i>=0; i--)
   cout<<a[i];
 }
```

Pointer to Pointer

```
void main()
{ int n=80;
  cout<<n;
  cout<<&n;
  int *pn;
  pn=&n;
  cout<<pn;
  cout<<&pn;
  cout<<*pn;
  int  **ppn;
  ppn=&pn;
  cout<<ppn;
  cout<<&ppn;
  cout<<*ppn;
  cout<<**ppn;
  }
```

Two Dimensional Arrays

- In C++

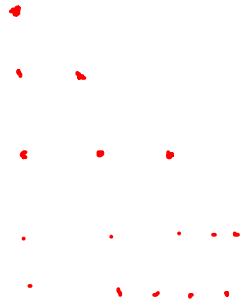
Static Array: `int  arr_2d[10][12];`

Dynamic Array:

```
  int **A;
A= new int*[rows];
 for (int i=0; i<rows; i++)
  A[i]=new int [cols];
```

Java arrays can be multidimensional. For example, a 2-dimensional array is an array of arrays. Two-dimensional arrays need not be rectangular. Each row can be a different length. Here's an example:

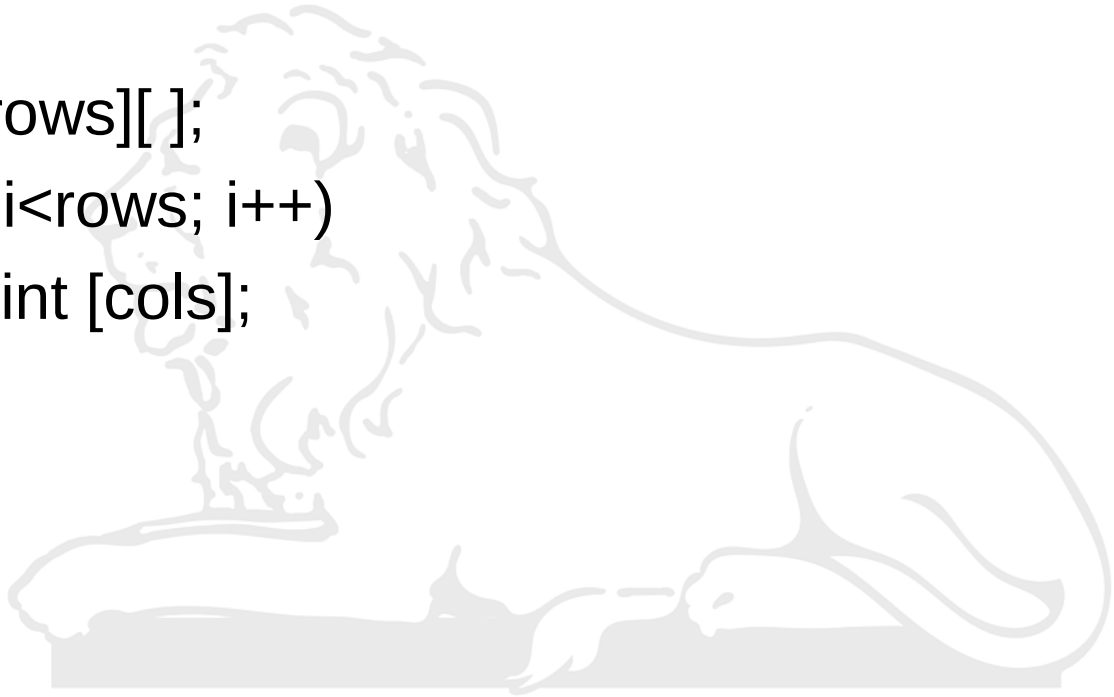
```
int [ ][ ] A;          // A is a two-dimensional array
A = new int[5][ ];     // A now has 5 rows, but no columns yet
A[0] = new int [1];  // A's first row has 1 column
A[1] = new int [2];  // A's second row has 2 columns
A[2] = new int [3];  // A's third row has 3 columns
A[3] = new int [5];  // A's fourth row has 5 columns
A[4] = new int [5];  // A's fifth row also has 5 columns
```



Two Dimensional Arrays

- In Java

```
int [ ] [ ] A;
A= new int[rows][ ];
 for (int i=0; i<rows; i++)
    A[i]=new int [cols];
```



In C++ Example 1

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

Output:

```
1   2   3
4   5   6
0   0   0
```

In C++, Example 2

```
int a[3][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
0   0   0   G9  G10 G11 G12
```

In C++,

Example 3

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

Output:

```
1    2    3
4    5    6
G1   G2   G3
```

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
```

In C++,

Example 5

```
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:

```
1    2    3
4    5    6
7    0    0
```

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