



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

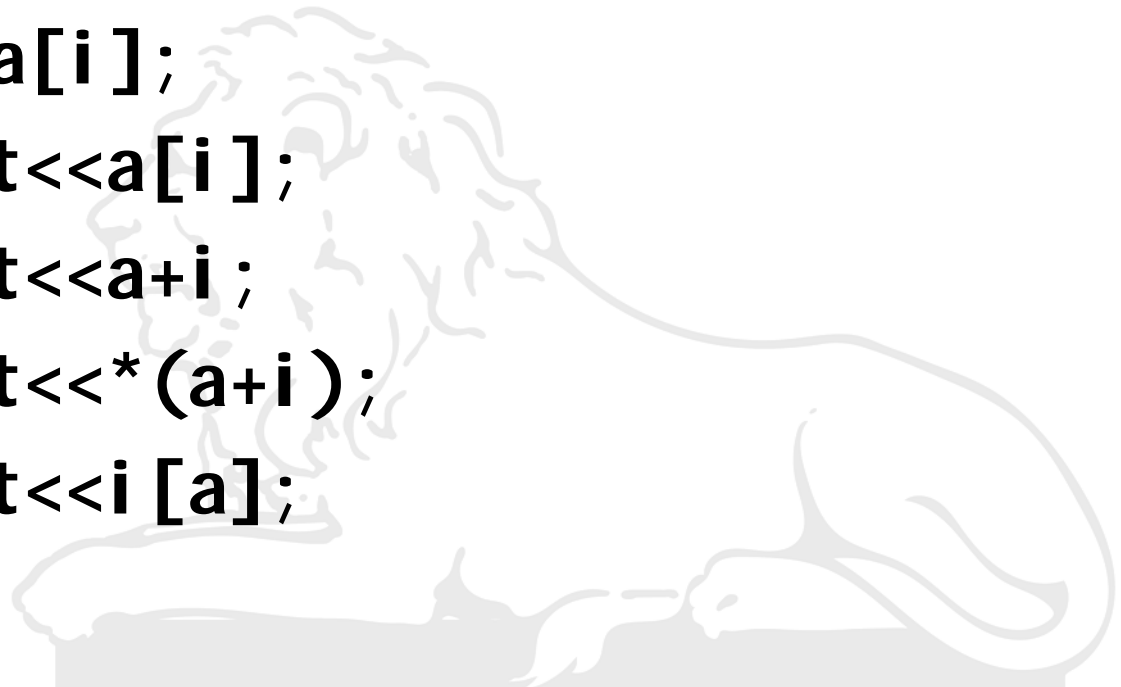
balarfcs@iitr.ac.in

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



Arrays and Pointers in C++

```
int a[10];  
for (int i=0; i<10; i++)  
{cin>>a[i];  
  cout<<a[i];  
  cout<<a+i;  
  cout<<*(a+i);  
  cout<<i[a];  
}
```

A faint, light-colored illustration of a lion statue, likely the Roorkee Lion, is visible in the background of the code block.

Pass by Value and Pass by Reference

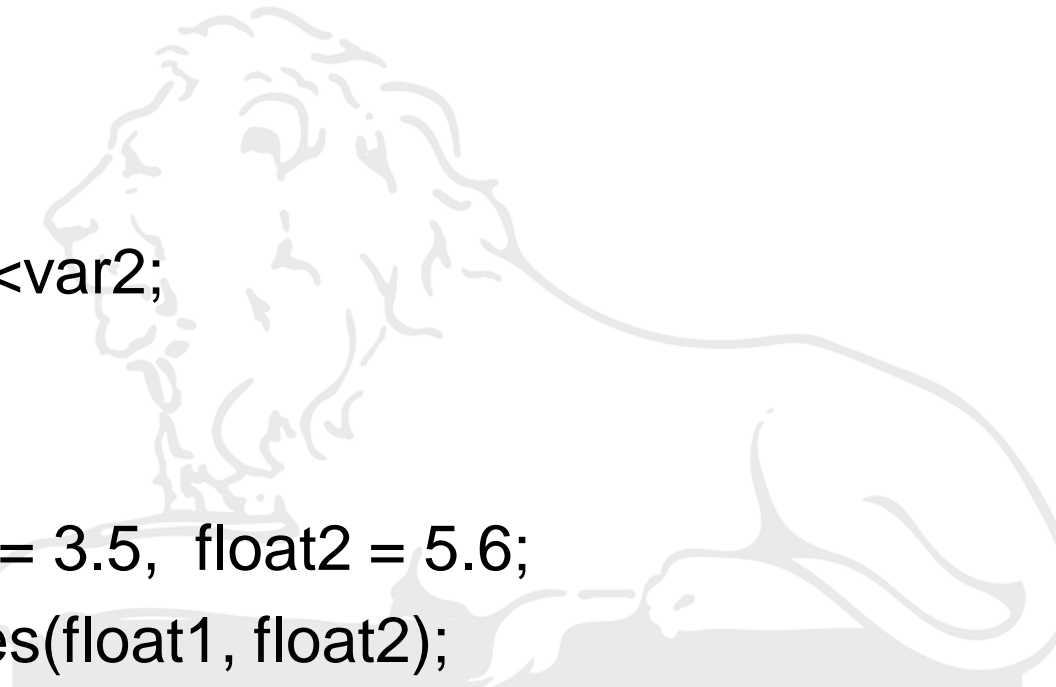
- We have seen Java is pass by Value
- Now we will take an example in C++ and see how it is pass by reference!



Pass by Value

```
void swapVariables(float var1, float var2)
{ float temp;
  temp = var1;
  var1 = var2;
  var2 = temp;
  cout<<var1<<var2;
}

int main( )
{ float float1 = 3.5, float2 = 5.6;
  swapVariables(float1, float2);
  cout<<float1<<float2;
  return 0;
}
```

A faint, stylized illustration of a lion lying down, positioned behind the code text.



```
1  #include <iostream>
2
3  using namespace std;
4
5  void swapVariables(float var1, float var2)
6  { float temp;
7    temp = var1;
8    var1 = var2;
9    var2 = temp;
10   cout<<"Inside Function"<<endl;
11   cout<<var1<<" "<<var2<<endl;
12  }
13
14  int main( )
15  { float float1 = 3.5, float2 = 5.6;
16    swapVariables(float1, float2);
17    cout<<float1<<" "<<float2<<endl;
18    return 0;
19  }
```

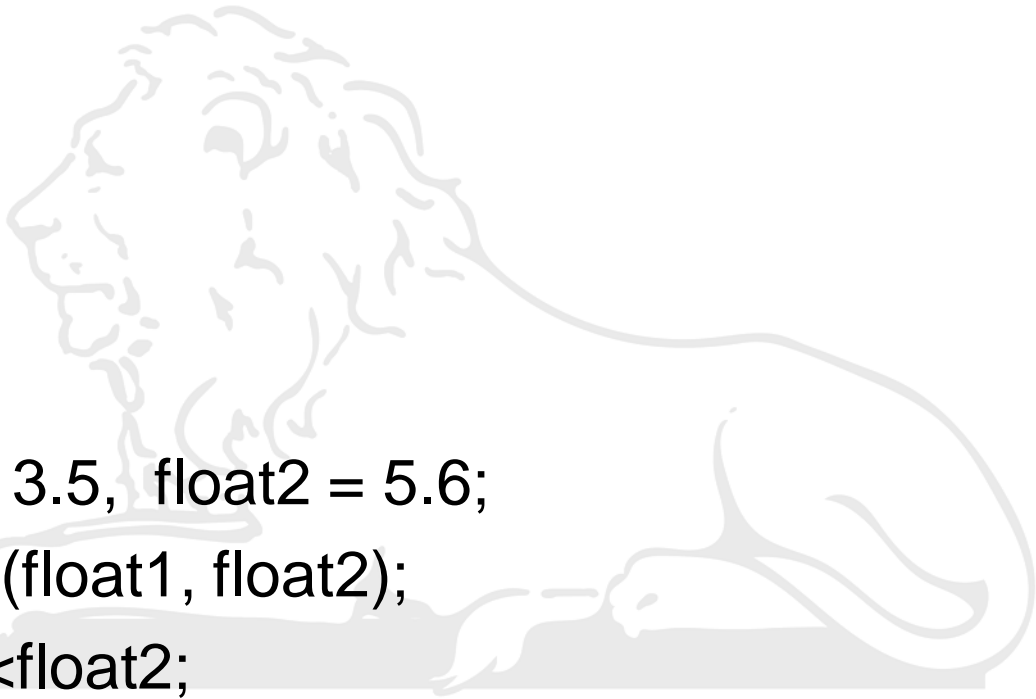
Terminal

```
sh-4.3$ g++ swapval.cpp
sh-4.3$ a.out
Inside Function
5.6 3.5
3.5 5.6
sh-4.3$
```

Pass by Reference

```
void swapVariables(float& var1, float& var2)
{ float temp;
  temp = var1;
  var1 = var2;
  var2 = temp;
}

int main( )
{ float float1 = 3.5, float2 = 5.6;
  swapVariables(float1, float2);
  cout<<float1<<float2;
  return 0;
}
```

A faint, light gray illustration of a lion lying down, facing left. The lion's head is turned slightly towards the viewer. The illustration is positioned behind the code text, serving as a background element.



```
1  #include <iostream>
2
3  using namespace std;
4
5  void swapVariables(float& var1, float& var2)
6  { float temp;
7    temp = var1;
8    var1 = var2;
9    var2 = temp;
10   cout<<"Inside Function"<<endl;
11   cout<<var1<<" "<<var2<<endl;
12  }
13
14  int main( )
15  { float float1 = 3.5, float2 = 5.6;
16    swapVariables(float1, float2);
17    cout<<"In Main"<<endl;
18    cout<<float1<<" "<<float2<<endl;
19    return 0;
20  }
```

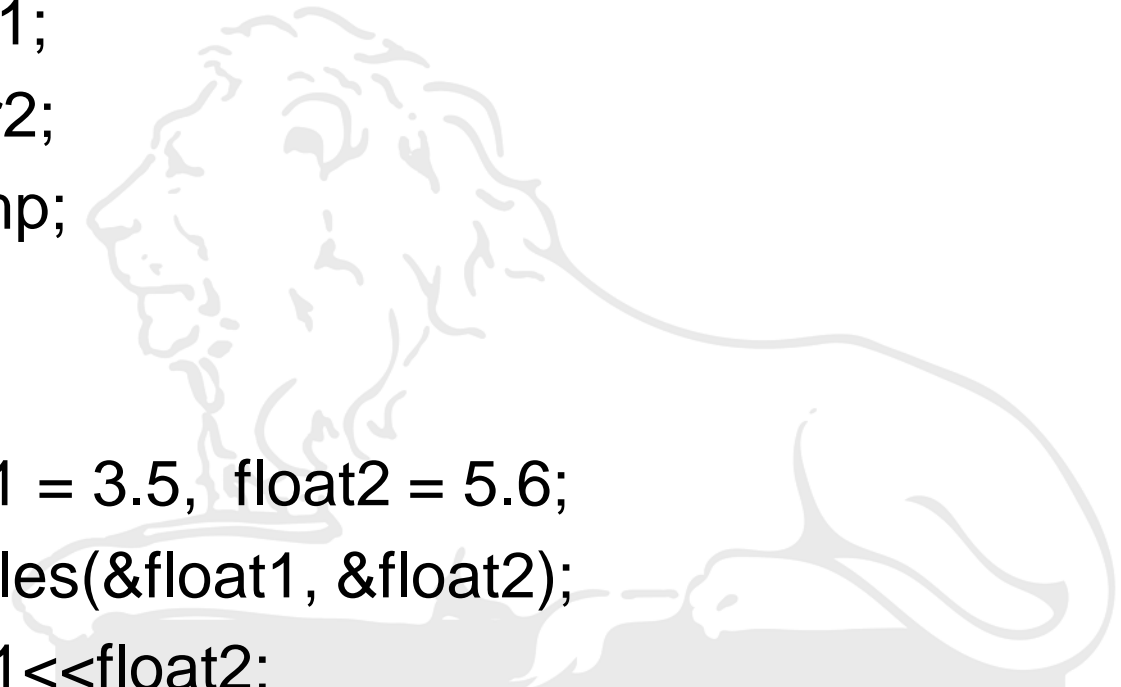
Terminal

```
sh-4.3$ g++ swapvar.cpp
sh-4.3$ a.out
Inside Function
5.6 3.5
In Main
5.6 3.5
sh-4.3$
```

Using pointers

```
void swapVariables(float* var1, float* var2)
{ float temp;
  temp = *var1;
  *var1 = *var2;
  *var2 = temp;
}

int main( )
{ float float1 = 3.5, float2 = 5.6;
  swapVariables(&float1, &float2);
  cout<<float1<<float2;
  return 0;
}
```

A faint, stylized illustration of a lion lying down, positioned behind the code text.



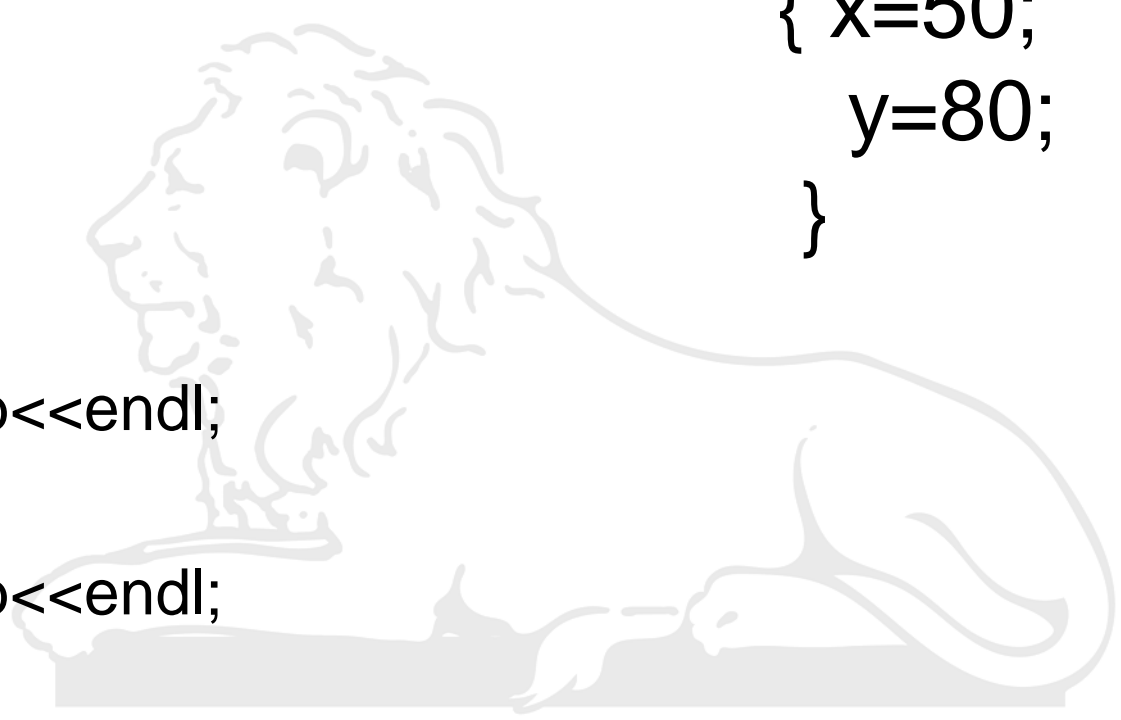
```
1  #include <iostream>
2
3  using namespace std;
4
5  void swapVariables(float* var1, float* var2)
6  { float temp;
7    temp = *var1;
8    *var1 = *var2;
9    *var2 = temp;
10   cout<<"Inside Function"<<endl;
11   cout<<*var1<<" "<<*var2<<endl;
12   }
13
14   int main( )
15   { float float1 = 31.5, float2 = 51.6;
16     swapVariables(&float1, &float2);
17     cout<<"In Main"<<endl;
18     cout<<float1<<" "<<float2<<endl;
19     return 0;
20   }
```

Terminal

```
sh-4.3$ g++ swappoint.cpp
sh-4.3$ a.out
Inside Function
51.6 31.5
In Main
51.6 31.5
sh-4.3$
```

```
void g(int x, int& y);  
void main()  
{ int a,b;  
a=20;  
b=15;  
g(a,b);  
cout<<a<<b<<endl;  
g(5*a-2,b);  
cout<<a<<b<<endl;  
g(a,5*b-3);  
cout<<a<<b<<endl;  
}
```

```
void g(int x, int& y)  
{ x=50;  
  y=80;  
}
```



Address of i^{th} location element in an Array

$\&a[i] = \&a[0] + i * e_size;$



Pointer and reference variables

`pv=0x7fffc9af9b88;`



`pv=0;`



`pv=NULL;`



`int v1=2*(*pv+v);`



`&w=2*(&u+&v);`



`int v2=2*(&u+&v);`



$pu+pv$



$*pu+*pv$



$pu=pu+1;$



$pv=*pu+pv;$



$pu=*pu+pv;$



$*pu=*pu+pu$