S J P N Trust's



HIRASUGAR INSTITUTE OF TECHNOLOGY, NIDASOSHI.

Inculcating Values, Promoting Prosperity

Approved by AICTE, Recognized by Govt. of Karnataka and Permanently Affiliated to VTU Belagavi.

Accredited at 'A' Grade by NAAC

Programmes Accredited by NBA: CSE, ECE, EEE & ME

Subject: Object Oriented Concepts (18CS45)

Module 2: Objects and Arrays (C++ Part)

Dr. Mahesh G Huddar

Dept. of Computer Science and Engineering

Arrays and Objects

- Like Structures, It is possible to have arrays of objects.
- The syntax for declaring and using an object array is exactly the same as it is for any other type of array.
- A array variable of type class is called as an array of objects.

#include<iostream>
using namespace std;

class Student

private: int rno; char name[20]; double avgmarks; public: void print_data()

cout<<"Roll No is "<<rno<<endl; cout<<"Name is "<<name<<endl; cout<<"Average marks "<<avgmarks<<endl;</pre>

void read_data() asoshi

cout<<"Enter the Roll No: "; cin>>rno; cout<<"Enter the name: "; cin>>name; cout<<"Enter average marks: "; cin>>avgmarks;

int main()

```
Student std[10];
cout<<"Enter the student data:"<<endl;
for (int i=0;i<3;i++)
      std[i].read_data(); E, HIT, Nidasoshi
for (int i=0;i<3;i++)
      cout<<i+1<<" Student information is:"<<endl;
      std[i].print data();
return 0;
```

#include<iostream> using namespace std; class Student private: int rno; char name[20]; double avgmarks; public: void read_data(); void print data();

};

void Student::read_data()

```
cout<<"Enter the Roll No: ";
cin>>rno;
cout<<"Enter the name: ";
cin>>name;
cout<<"Enter the average marks: ";
cin>>avgmarks;
```

void Student::print_data()

cout<<"Roll No is "<<rno<<endl; cout<<"Name is "<<name<<endl; cout<<"Average marks "<<avgmarks<<endl;</pre>

```
int main()
      Student std[10];
      cout<<"Enter the student data:"<<endl;
     for (int i=0;i<3;i++)
           std[i].read_data();, HIT, Nidasoshi
      for (int i=0;i<3;i++)
            cout<<i+1<<" Student information is:"<<endl;
            std[i].print_data();
      return 0;
```

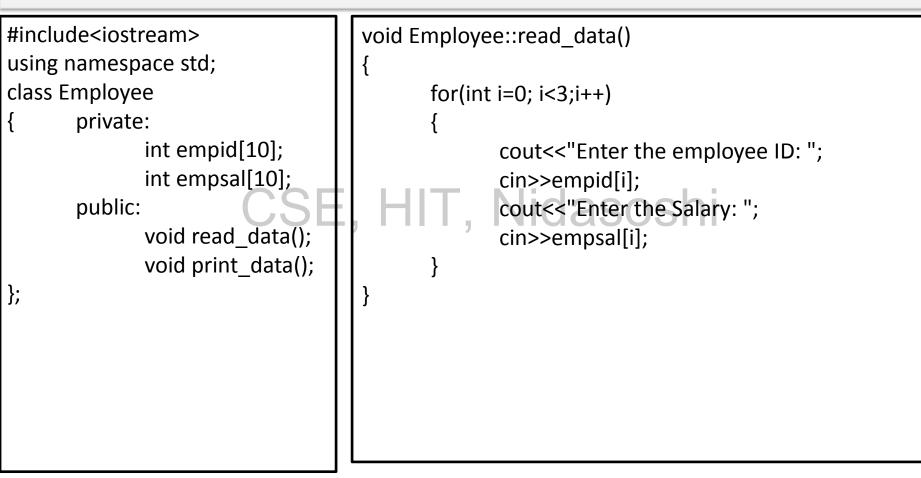
Array inside objects

- An array can be used as a member variable of class.
- Using the instance or object of that class we can initialize the array



elements.

Array inside objects - Example



Array inside objects - Example

```
void Employee::print_data()
```

```
for (int i=0;i<3;i++)
{
     cout<<"Employee "<<i+1<<" ID is "<< empid[i]<<endl;
     cout<<"Employee "<<i+1<<" Salary is "<< empsal[i]<<endl;
}</pre>
```

```
int main()
```

Employee emp; emp.read_data(); emp.print_data(); return 0;

Namespace

- Namespace is a new concept introduced by the ANSI C++ standards committee.
- For using identifiers it can be defined in the namespace scope as below.
- Syntax: CSE, HIT, Nidasoshi
 - using namespace std;
 - In the above syntax "std" is the namespace where ANSI C++ standard class libraries are defined.
 - Even own namespaces can be defined.



Syntax:

namespace namespace_name

CSE, HIT, Nidasoshi //Declaration of variables, functions, classes, etc.

Namespace - Example

```
#include<iostream>
using namespace std;
namespace nsl
     int a = 10;
              CSE, HIT, Nidasoshi
namespace ns2
     float a = 20.5;
int main()
     cout<<"Namespace example"<<endl;</pre>
     cout<<"The value of a in ns1 is "<<ns1::a<<endl;</pre>
     cout<<"The value of a in ns2 is "<<ns2::a<<endl;
     return 0;
```

We will learn:

- What are Nested Classes...?
- How to define nested class as private member of enclosing class?
- How to define nested class as public member of enclosing class?
- Define member functions of Nested class outside the enclosing class?
- How to define Nested class outside the enclosing class?

Definition of Nested Class:

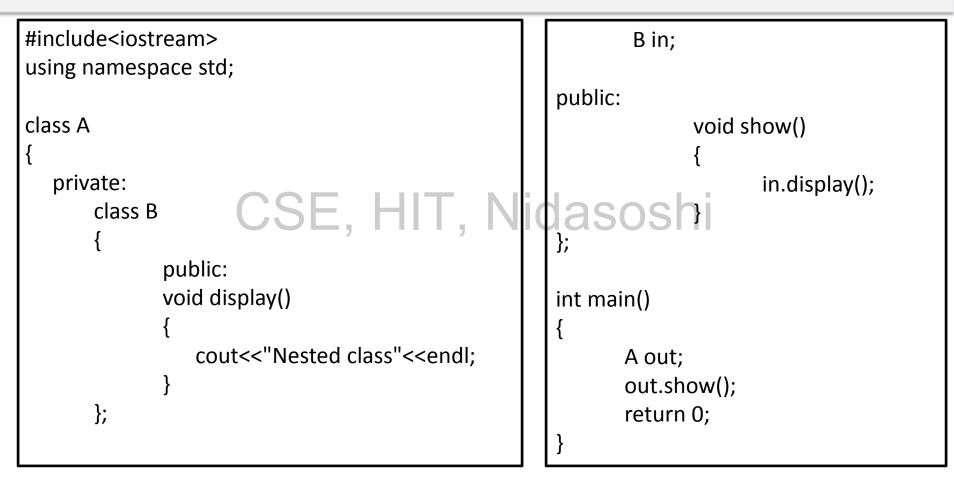
• Nested class is a class defined inside another class. The class which

contains the nested class is called as **Enclosing class**.

How to define nested class as private member of enclosing class?

- The nested class can be defined as private member of enclosing CSE, HIT, Nidasoshi
- The object of enclosing class can be used to access the data

members and member functions of the nested class.



How to define nested class as public member of enclosing class?

- We can define the nested class as a public member of enclosing CSE, HIT, Nidasoshi class.
- In this case, the public member function of nested class can be

accessed from the object of enclosing class directly.

<pre>#include<iostream> using namespace std:</iostream></pre>	
using namespace std;	
class A	int main()
{	{
public:	A::B in;
class B	in.display();
CSE. HIT	Nireturn 0; Shi
public:	}
void display()	
{	
cout<<"Nested class"< <endl;< td=""><td></td></endl;<>	
}	
};	
};	

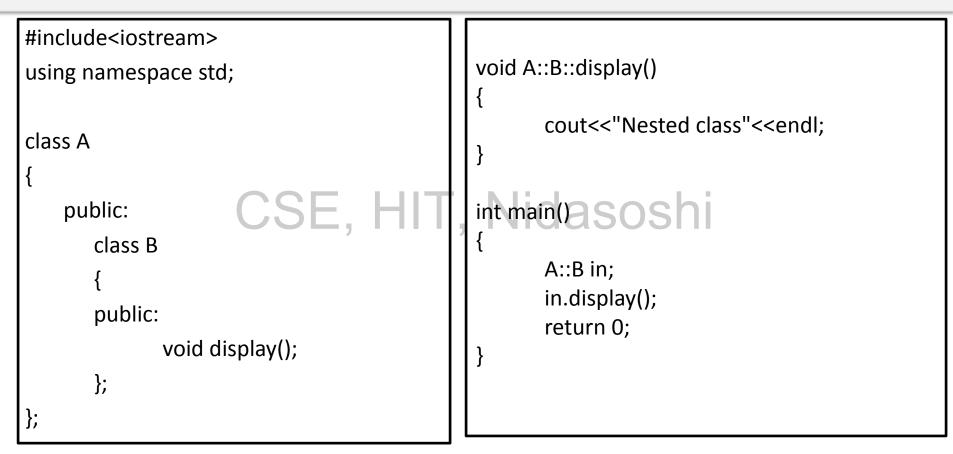
How to define member functions of Nested class outside the

enclosing class?

• The member function of the nested class can be defined outside

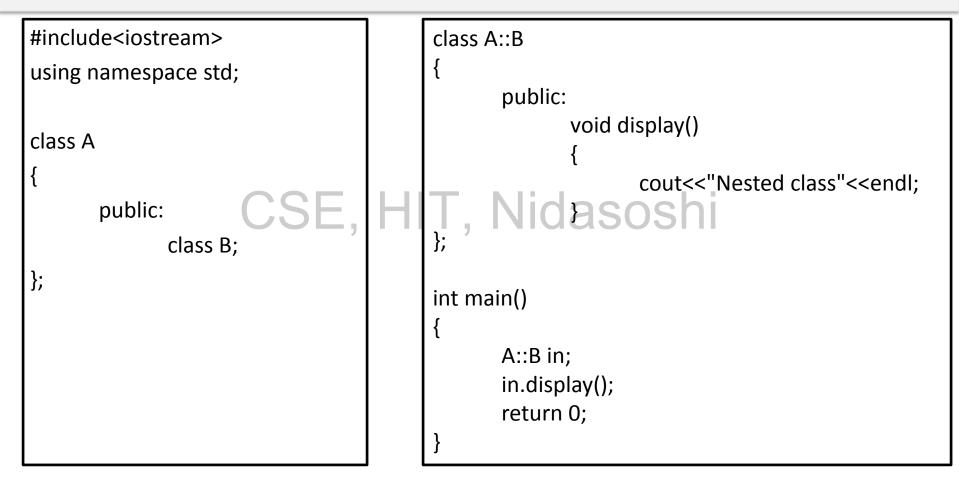
the enclosing class.

• Here we need to use the scope resolution operator.



How to define Nested class outside the enclosing class?

- The nested class can be defined outside the enclosing class.
- Here we need to use the scope resolution operator.



Constructors in C++

We will Learn:

- What are Constructors?
- What are the types of constructors with simple programming example?

Constructors in C++

Constructors are special class functions which performs initialization of every object.

The Compiler calls the Constructor whenever an object is created.

Constructor's initialize values to data members after storage is allocated to the object.

While defining a constructor you must remember that the name of constructor will be same as the name of the class, and constructors never have return type.

Constructors can be defined either inside the class definition or outside class definition using class name and scope resolution :: operator.

Type of Constructor in C++

Constructors are of four types :

Default Constructor

Parameterized Constructor CSE, HIT, Nidasoshi

Copy Constructor

Explicit Constructor

Default Constructor in C++

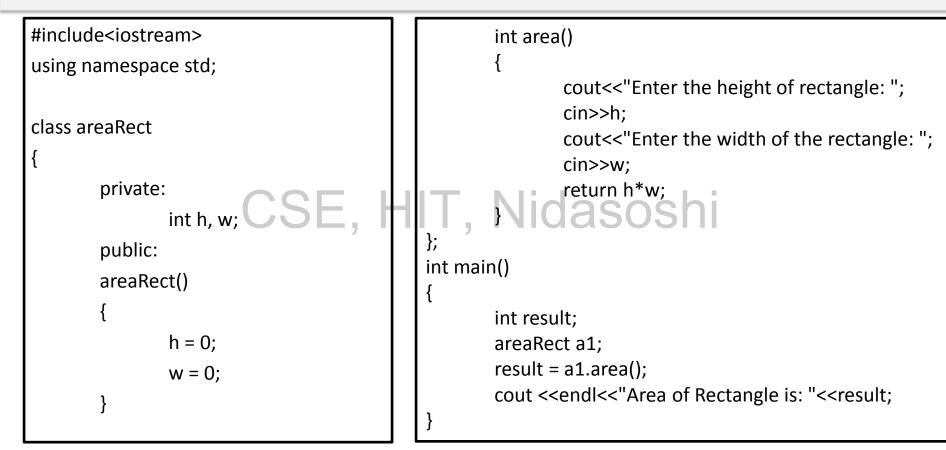
Default constructor is the constructor which doesn't take any argument.

It has no parameters

Syntax : class_name ()SE, HIT, Nidasoshi

Constructor Definition

Default Constructor in C++



Parameterized Constructor in C++

- These are the constructors with parameter.
- Using this Constructor you can provide different values to data members of different objects, by passing the appropriate values as

argument.

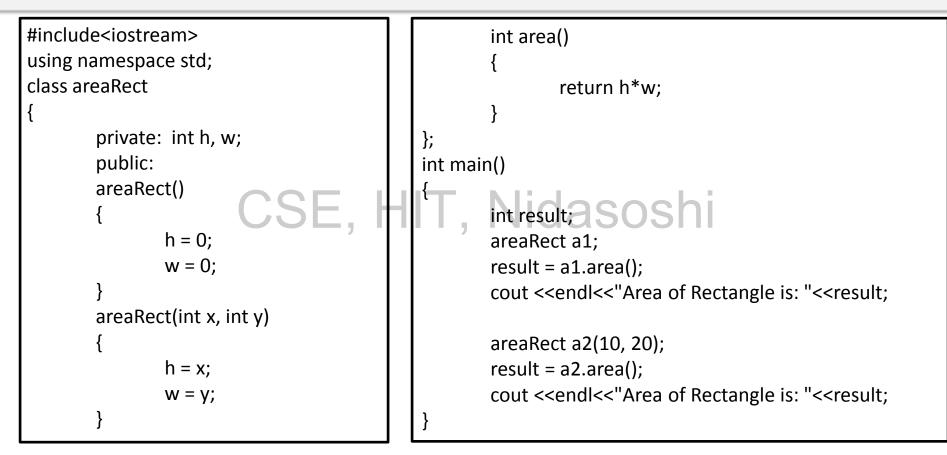
Parameterized Constructor in C++

• Syntax :

}

class_name (parameters)	
{ Constructor Definition	Nidasoshi

Parameterized Constructor in C++



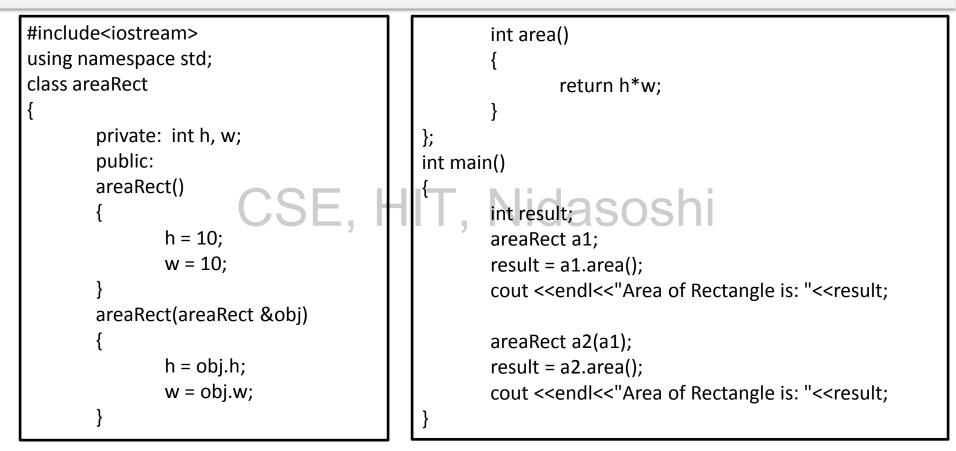
Copy Constructor in C++

Copy constructor is a special type of constructor in which new object is created as a copy of an existing object.

The copy constructor is called whenever a new variable is created from an object.



Copy Constructor in C++



Explicit Constructor in C++

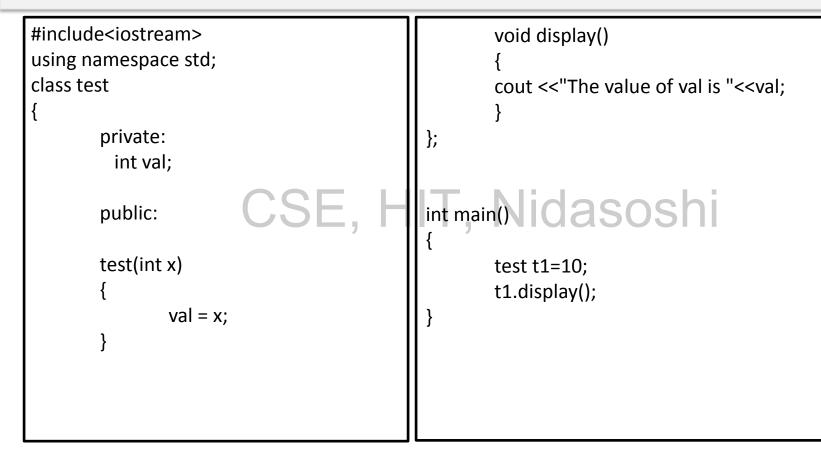
In C++, compiler is allowed to make one time type conversion to resolve the parameters to functions.

In C++, a constructor with only one required parameter is considered as an implicit conversion function. It converts the parameter type to the class type.

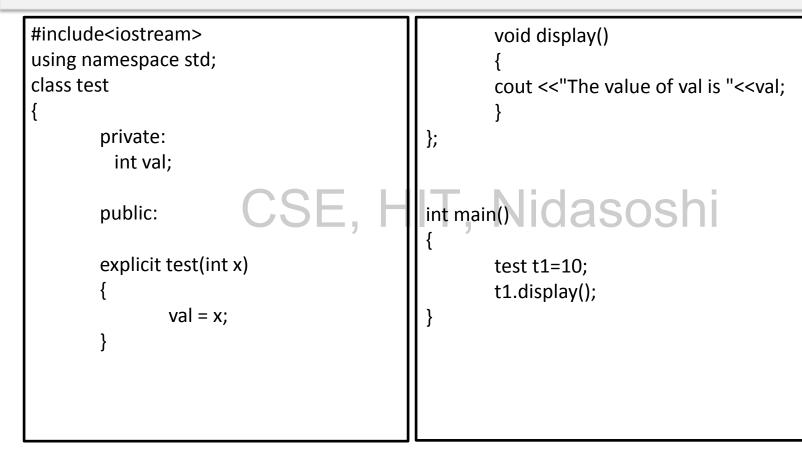
Prefixing explicit keyword prevents the constructor from using the implicit conversion. Syntax :

```
explicit class_name (parameters)
{
Constructor Definition
```

Explicit Constructor in C++



Explicit Constructor in C++



Constructor Overloading in C++

- Just like other member functions, constructors can also be overloaded.
- In fact when you have both default and parameterized constructors defined in your class you are having Overloaded Constructors, one with no parameter and other with parameter.
- You can have any number of Constructors in a class that differ in parameter list.

Destructors in C++

- Destructor is a special class function which destroys the object as soon as the scope of object ends.
- The destructor is called automatically by the compiler when the object goes out of scope. HIT, Nidasoshi
- The syntax for destructor is same as that for the constructor, the class name is used for the name of destructor, with a tilde ~ sign as prefix to it.