

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

## Constructors for classes

Comp Sci 1570 Introduction to C++



Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

# C++ supports several basic ways to initialize

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object  
Copy constructor  
Operator=

Classes in  
classes

```
int nValue; // declare but not define  
nValue = 5; // assign
```

```
int nValue = 5; // copy initialization
```

```
int nValue(5); // direct initialization
```

```
int value{5}; // uniform init. (C++ 11 only)
```

```
MyClass my_object; // declare but not define  
// initialization?
```

Even though direct initialization form looks a lot like a function call, the compiler keeps track of which names are variables and which are functions. Direct initialization can perform better than copy initialization for some data types. It also helps differentiate initialization from assignment.

Initialization  
review

**Definitions**

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

- 1 Initialization review
- 2 **Definitions**
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object  
Copy constructor  
Operator=

Classes in  
classes

To be able to initialize your user-defined types, you must write a special kind of member function for your class. It is called a constructor. Constructors are really rather special in that they have some key features:

- Constructors are always named the name of the class (with the same capitalization)
- Constructors have no return type (not even void) and hence has no return statement.
- Constructors can be overloaded like any other function, and you will most likely do so.
- Constructors are called by the compiler automatically; they are rarely called explicitly by the programmer.
- If you write no constructor, the compiler will provide your class with a default constructor. The mechanism is suppressed once you write any constructor.

Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

**Default**

Parameterized

Copy

Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor

Operator=

Classes in  
classes

- 1 Initialization review
- 2 Definitions
- 3 **Types of constructor**
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

**Default**  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

A constructor that accepts no parameters is known as default constructor. If no constructor is defined then the compiler supplies a default constructor.

...

```
Circle :: Circle()
{
    radius = 0;
}
```

```
int main(){
    Circle my_circle;
    my_circle.radius=4;
}
```

...



Initialization review

Definitions

Types of constructor

Default  
**Parameterized**  
 Copy  
 Destructors

Overloading constructors

Member initialization lists

Initialize from existing object  
 Copy constructor  
 Operator=

Classes in classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized**
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

Default  
**Parameterized**  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

A constructor that receives arguments/parameters, is called parameterized constructor.

...

```
Circle :: Circle(double r)
{
    radius = r;
}
```

```
int main(){
    Circle my_circle(4);
```

...

Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
**Copy**  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy**
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructorDefault  
Parameterized  
**Copy**  
DestructorsOverloading  
constructorsMember  
initialization  
listsInitialize from  
existing objectCopy constructor  
Operator=Classes in  
classes

A constructor that initializes an object using values of another object passed to it as parameter, is called copy constructor. It creates the copy of the passed object.

...

```
Circle :: Circle(Circle &otherCircle)
{
    radius = otherCircle.radius;
}
```

```
int main(){
    Circle my_circle(4);
    Circle my_circle2(my_circle);
```

...

Initialization review

Definitions

Types of constructor

Default

Parameterized

Copy

**Destructors**

Overloading constructors

Member initialization lists

Initialize from existing object

Copy constructor

Operator=

Classes in classes

- 1 Initialization review
- 2 Definitions
- 3 **Types of constructor**
  - Default
  - Parameterized
  - Copy
  - Destructors**
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy

**Destructors**

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

A destructor is a member function having same name as that of its class preceded by (tilde) sign and which is used to destroy the objects that have been created by a constructor. It gets invoked when an object's scope is over.

```
~Circle () {}
```

We don't cover these this semester

Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
 existing object  
 Copy constructor  
 Operator=

Classes in  
classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

Like any other function, a constructor can also be overloaded with different versions taking different parameters: with a different number of parameters and/or parameters of different types. The compiler will automatically call the one whose parameters match the arguments



Initialization review

Definitions

Types of constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading constructors

Member initialization lists

Initialize from existing object

Copy constructor  
 Operator=

Classes in classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization review

Definitions

Types of constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading constructors

Member initialization lists

Initialize from existing object

Copy constructor  
 Operator=

Classes in classes

- When a constructor is used to initialize other members, these other members can be initialized directly, without resorting to statements in its body, by inserting, before the constructor's body, a colon (:) and a list of initializations for class members.
- For members that cannot be default-initialized, such as members of reference and const-qualified types, member initializers must be specified.
  - If non-static const data members in your class have default constructors and you don't use a constructor initializer list, you won't be able to initialize them to intended state as they will be initialized to their default state.
  - Reference data members must be initialized when compiler enters constructor as references can't be just declared & initialized later. This is possible only with constructor initializer list.
- Efficiency: rather than calling default init just to overwrite later, initialize directly

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

```
class Rectangle
{
    int width , height ;
    public :
        Rectangle(int , int );
        int area () { return width*height ; }
};
```

```
Rectangle :: Rectangle (int x , int y)
{ width=x ; height=y ; }
// or
Rectangle :: Rectangle (int x , int y) : width (x)
{ height=y ; }
// or
Rectangle :: Rectangle (int x , int y) :
width (x) , height (y)
{ }
```

Initialization review

Definitions

Types of constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading constructors

Member initialization lists

Initialize from existing object

Copy constructor  
 Operator=

Classes in classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization review

Definitions

Types of constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading constructors

Member initialization lists

Initialize from existing object

Copy constructor  
 Operator=

Classes in classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

```

class MyClass
{
    public :
        int a, b; string c;
};
    
```

```

MyClass::MyClass(const MyClass& existingObj):
a(x.a), b(x.b), c(x.c) {}
    
```

```

int main(){
    MyClass
    
```

...

Initialization review

Definitions

Types of constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading constructors

Member initialization lists

Initialize from existing object

Copy constructor  
 Operator=

Classes in classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

# Operator= provided by default (copy assignment)

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

```
MyClass foo;
```

```
// object initialization: copy constructor  
MyClass bar (foo);
```

```
// object initialization: copy constructor  
MyClass baz = foo;
```

```
// object already initialized: copy assignment  
foo = bar;
```

Body of overloading function is similar to copy constructor



Initialization  
review

Definitions

Types of  
constructor

Default  
 Parameterized  
 Copy  
 Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
 Operator=

Classes in  
classes

- 1 Initialization review
- 2 Definitions
- 3 Types of constructor
  - Default
  - Parameterized
  - Copy
  - Destructors
- 4 Overloading constructors
- 5 Member initialization lists
- 6 Initialize from existing object
  - Copy constructor
  - Operator=
- 7 Classes in classes

Initialization  
review

Definitions

Types of  
constructor

Default  
Parameterized  
Copy  
Destructors

Overloading  
constructors

Member  
initialization  
lists

Initialize from  
existing object

Copy constructor  
Operator=

Classes in  
classes

- A class may contain other classes as member variables.
- By default, when the outer class is constructed, the member variables will have their default constructors called.
- This happens before the body of the constructor executes.