Object-Oriented Programming Using C++

Constructor and Destructor

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Constructor

- A class constructor is a special member function of a class that is executed whenever we create new objects of that class
- A constructor will have exact same name as the class and it does not have any return type at all, not even void
- A default constructor does not have any parameter, but if you need, a constructor can have parameters
  - This helps you to assign initial value to an object at the time of its creation

Example: Constructor

Constructor for the Line class

```
class Line {
   private:
      double length;
   public:
      void setLength( double len );
      double getLength( void );
      Line();  // This is the constructor
};

// Member functions definitions including constructor
Line::Line(void) {
   cout << "Object is being created" << endl;
}
void Line::setLength( double len ) {
   length = len;
}
double Line::getLength( void ) {
   return length;
}

int main() {
   Line line;
   // set line length
   line.setLength(6.0);
   cout << "Length of line : " << line.getLength() <<endl;
   return 0;
}
```

Example: Constructor

Output

```
Object is being created
Length of line : 6
```

Example: Constructor with Parameter

Constructor with parameter for the Line class

```
class Line {
   private:
      double length;
   public:
      void setLength( double len );
      double getLength( void );
      Line(double len);  // This is the constructor
};

// Member functions definitions including constructor
Line::Line( double len) {
   cout << "Object is being created, length = " << len << endl;
   length = len;
}
void Line::setLength( double len ) {
   length = len;
}
double Line::getLength( void ) {
   return length;
}

int main() {
   Line line(10.0);
   cout << "Length of line : " << line.getLength() << endl;
   line.setLength(6.0);    // set line length again
   cout << "Length of line : " << line.getLength() << endl;
   return 0;
}
```

Example: Constructor with Parameter

Output

```
Object is being created, length = 10
Length of line : 10
Length of line : 6
```

Destructor

- A special member function of a class that is executed whenever an object of its class goes out of scope
- Has exact same name as the class prefixed with a tilde ($\sim$)
- Can neither return a value nor can it take any parameters
- Very useful for releasing resources before coming out of the program like closing files, releasing memories etc.

Example: Destructor

Destructor for the Line class

```
class Line
   private:
      double length;
   public:
      void setLength( double len );
      double getLength( void );
      Line();   // This is the constructor declaration
      ~Line();  // This is the destructor: declaration
};

Line::Line(void) {
   cout << "Object is being created" << endl;
}
Line::~Line(void) {
   cout << "Object is being deleted" << endl;
}
void Line::setLength( double len ) {
   length = len;
}
double Line::getLength( void ) {
   return length;
}

int main() {
   Line line;
   line.setLength(6.0);
   cout << "Length of line : " << line.getLength() <<endl;
   return 0;
}
```

Example: Destructor

Output

```
Object is being created
Length of line : 6
Object is being deleted
```

Object-Oriented Programming Using C++

Constructor and Destructor

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur