

Initializing Variables in a Class Hierarchy



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

By the end of this video you will be able to...

- Use same-class and super class constructors in class creation

```
Student s = new Student();
```



Student

Person

Object

```
public class Student extends Person
{
}
```



```
public class Student extends Person
{
    public Student()
    {
        super();
    }
}
```

**But how do we
initialize name ?**


```
public class Person extends Object
{
    private String name;
    public Person() {
        super();
    }
}
```

Initialize name
variable

```
public class Person extends Object
{
    private String name;

    public Person( String n ) {
        this.name = n;
        super();
    }
}
```

```
public class Person extends Object
{
    private String name;
    public Person( String n ) {
        this.name = n;
        super();
    }
}
```



ERROR!

**super () has to be the
first line!**

```
public class Person extends Object
{
    private String name;

    public Person( String n ) {
        super();
        this.name = n;
    }
}
```

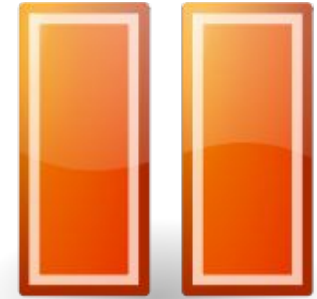


```
public class Person extends Object
{
    private String name;
    public Person( String n ) {
        super();
        this.name = n;
    }
}
```

Initialize name
variable in
student

```
public class Student extends Person
{
    public Student()
    {
        super();
    }
}
```

```
public class Person extends Object
{
    private String name;
    public Person( String n ) {
        super();
        this.name = n;
    }
}
```



```
public class Student extends Person
{
    public Student( String n )
    {
        super();
        this.name = n;
    }
}
```

```
public class Person extends Object
{
    private String name;
    public Person( String n ) {
        super();
        this.name = n;
    }
}
```

```
public class Student extends Person
{
    public Student( String n )
    {
        super();
        this.name = n;
        super(n) ;
    }
}
```

```
public class Student extends Person
{
    public Student( String n )
    {
        super(n) ;
    }
}
```

Let's add a no-
arg
constructor

```
public class Student extends Person
{
    public Student( String n )
    {
        super(n) ;
    }

    public Student ()
    {
        super( "Student" ) ;
    }
}
```

```
public class Student extends Person
{
    public Student( String n )
    {
        super (n) ;
    }

    public Student ()
    {
        this( "Student" ) ;
    }
}
```

**Use our same class
constructor**



