Writing Constructors

- The constructor of a class is automatically executed when an object is instantiated
- Variables should be initialized in the constructor
- General form

```
Public <class name>(<parameters>) {
      <statements>
}
```

no return type
always has same name
as class

- Constructors **can** be overloaded to provide more options
    - For example, overload the Circle constructor to include a radius
    - Just like with an overloaded method, the compiler uses the number and types of parameters to determine which constructor to execute

Add this to the Circle class to overload the constructor	Now you can create an object with this statement:

`/*`
` * constructor`
` * pre: none`
` * post: A Circle object created with radius r.`
` */`
`Public Circle(double r) {`
`    radius = r;`
`}`
	`Circle spot = new Circle(5);`

Programming Exercise:

a) Modify the Circle class to include an overloaded constructor that accepts the radius of the Circle object, as shown in the previous section.

b) Design and then create a Rectangle class that has overloaded constructors. The first constructor requires no parameters. The second has two parameters, one for length, and a second for width. Member variables store the length and width of the rectangle, and member methods assign and retrieve the length and width and return the area and perimeter of the rectangle. Test the class by writing appropriate client code.

Do not submit your code for either part a or part b just yet.