Overloading and Overriding Methods
Java permits a class to replace (or override) a method that it has it has inherited. A subclass can define a method with the SAME signature as a superclass method.

- **Overloading**
  - Two methods have the same name but DIFFERENT signatures -- the parameter types and/or number of parameters are different.
    - public StringBuilder append (int a);
    - public StringBuilder append (String s);
    - public StringBuilder append (int a, int b);

- **Overriding**
  - A method in the subclass has the SAME signature as a method in the superclass.

Calling An Overridden Method
Within a subclass definition, you can call the superclass method with the “super” reference.

```
public class HourlyWorker extends Employee {
    public String toString() {
        //explicitly calls Employee’s toString method
        return super.toString() + “Hours: “ + hours + “ Rate: “ + rate;
    }
}
```

1. Under what circumstances would a subclass need to override a superclass method?

2. If a method in a subclass has the same name as a method in the superclass, but uses a different parameter list, does the subclass method overload or override the superclass method?

3. How do you prevent a method from being overridden?

4. A superclass has the following method:
   ```
   public void setValue (int b)
   {
       value = v;
   }
   ```
   Write a statement that may appear in a subclass that calls this method, passing 10 as an argument.