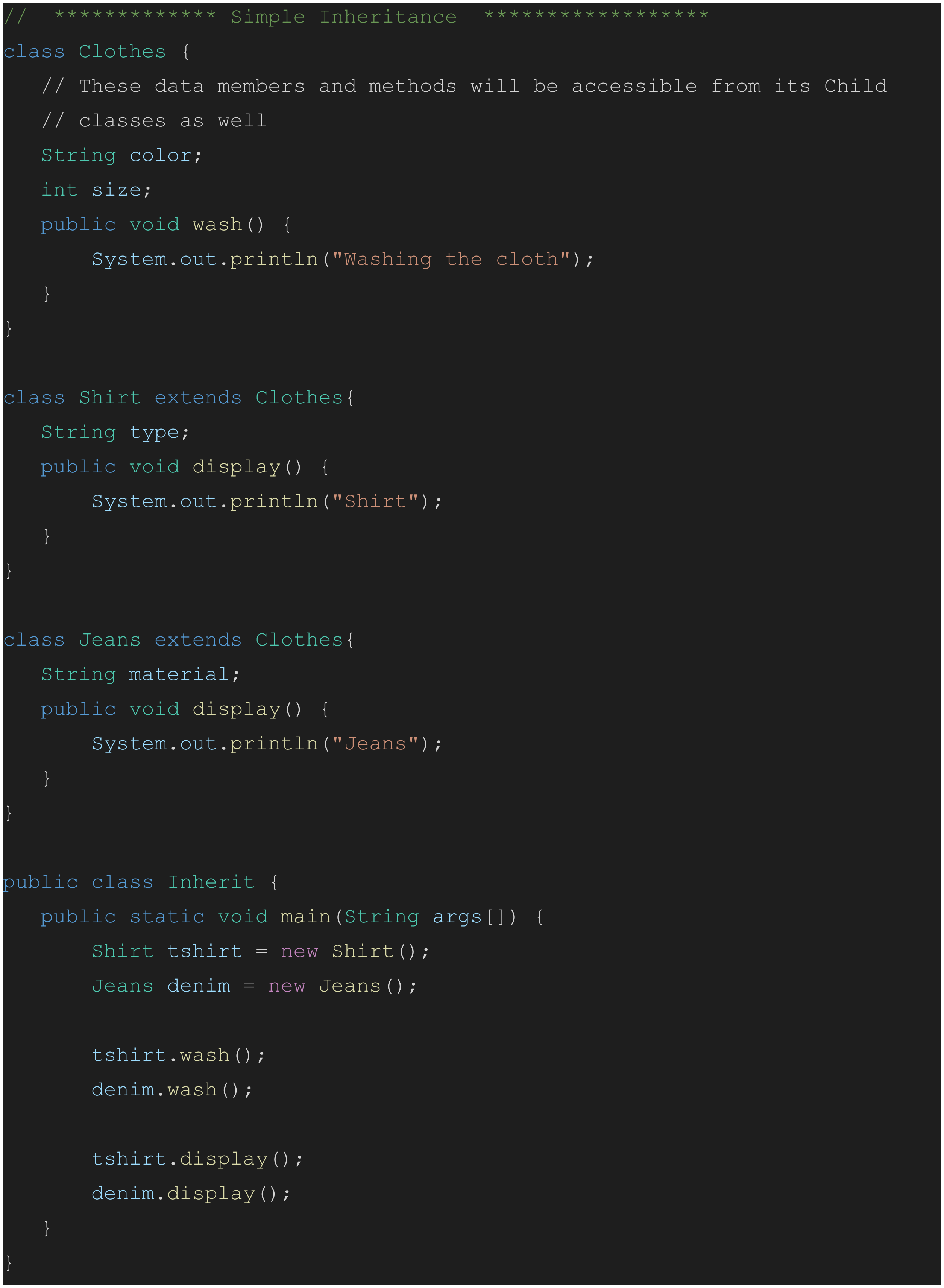
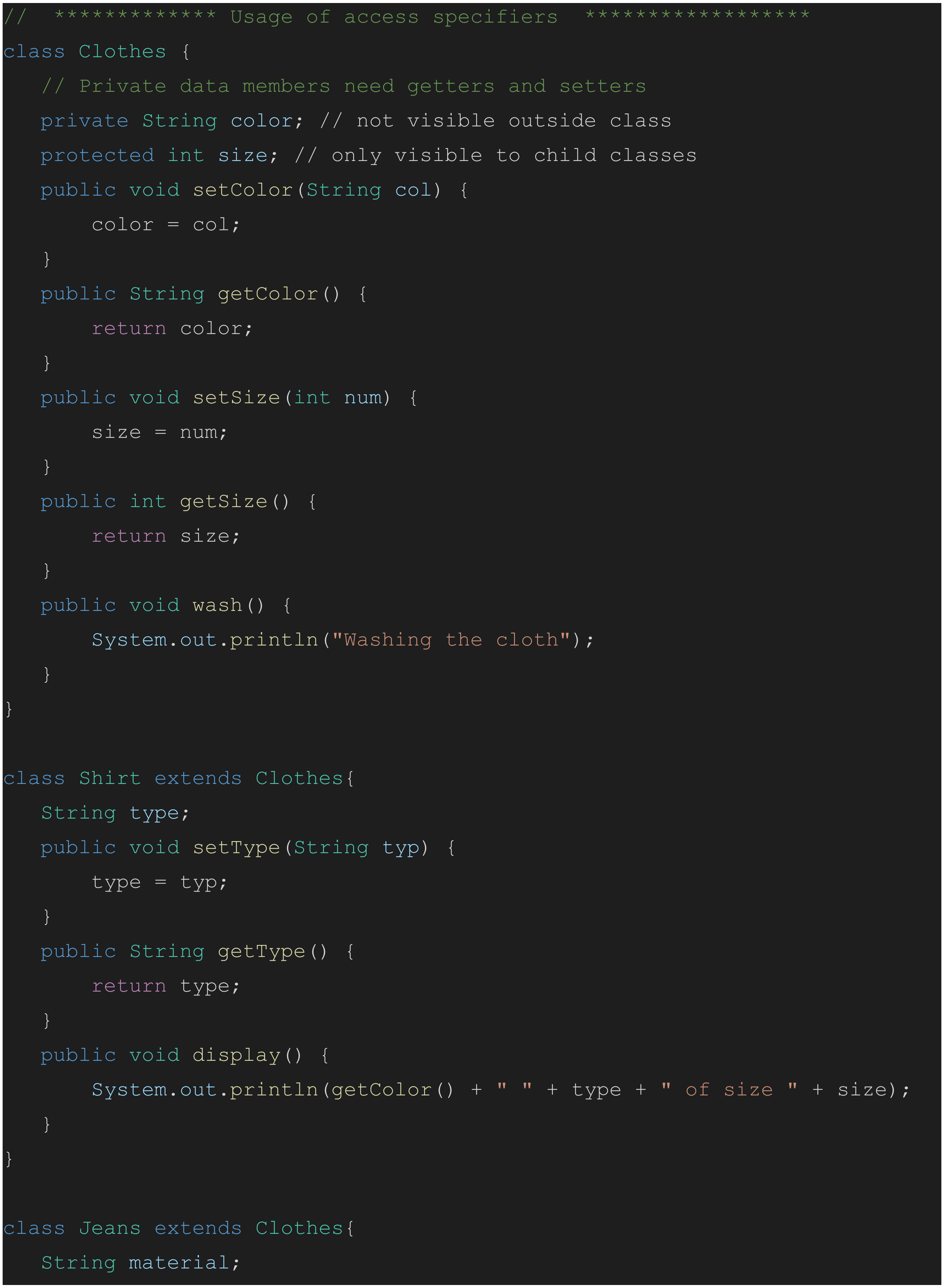
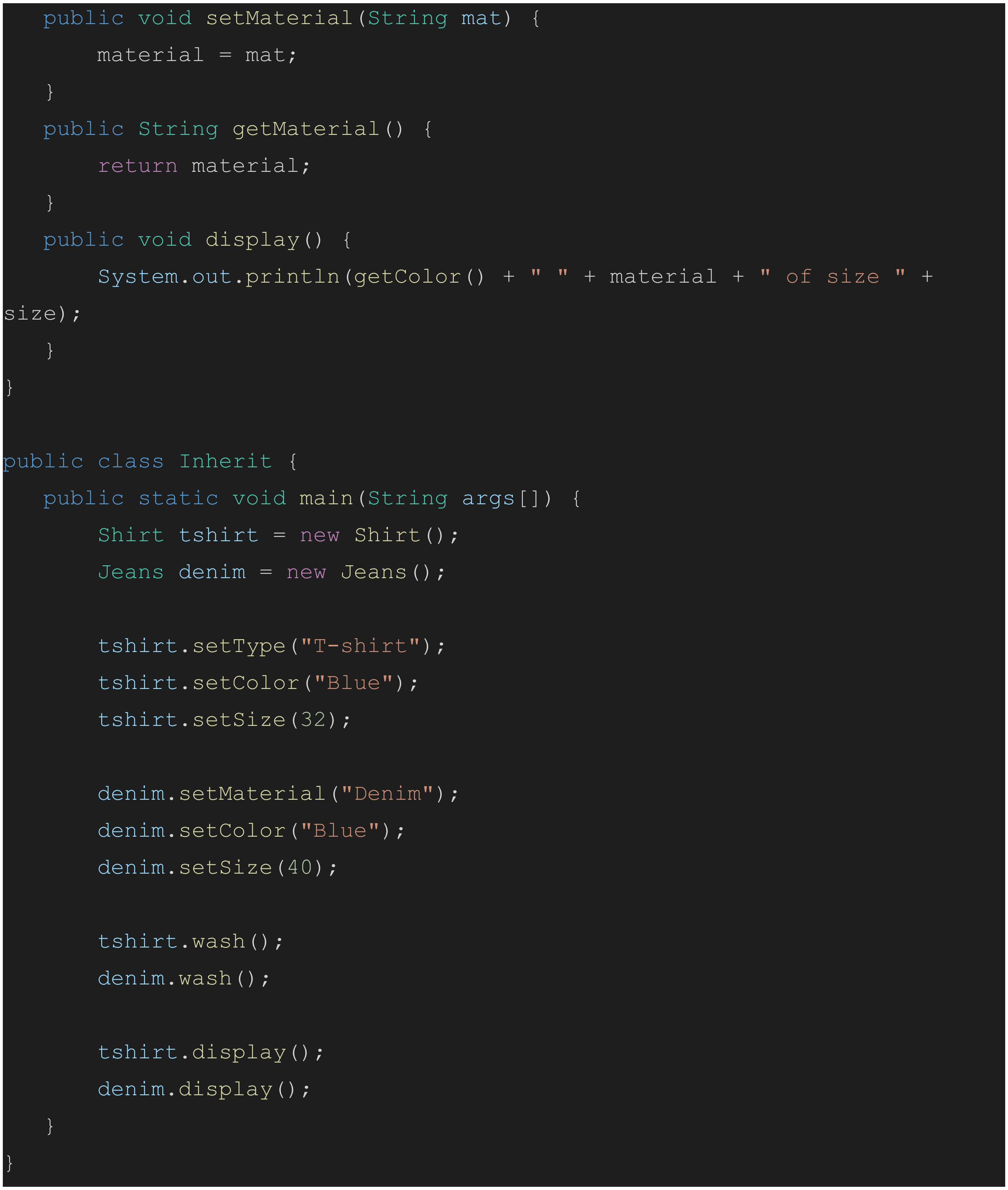


```
//  ************* Simple Inheritance  ******************
class Clothes {
   // These data members and methods will be accessible from its Child
   // classes as well
   String color;
   int size;
   public void wash() {
       System.out.println("Washing the cloth");
   }
}

class Shirt extends Clothes{
   String type;
   public void display() {
       System.out.println("Shirt");
   }
}

class Jeans extends Clothes{
   String material;
   public void display() {
       System.out.println("Jeans");
   }
}

public class Inherit {
   public static void main(String args[]) {
       Shirt tshirt = new Shirt();
       Jeans denim = new Jeans();

       tshirt.wash();
       denim.wash();

       tshirt.display();
       denim.display();
   }
}
```



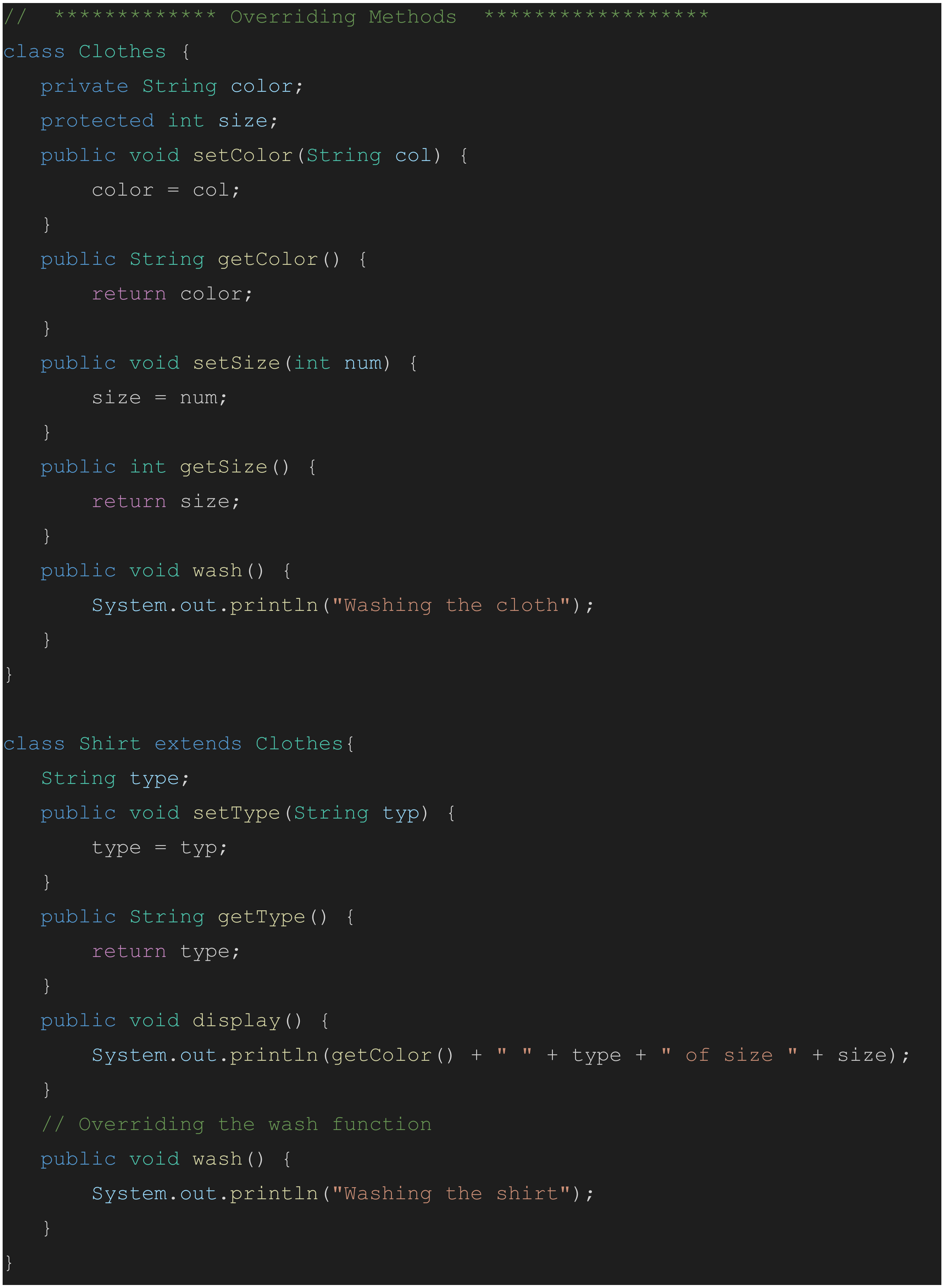
```java
//  ************* Usage of access specifiers  ******************
class Clothes {
   // Private data members need getters and setters
   private String color; // not visible outside class
   protected int size; // only visible to child classes
   public void setColor(String col) {
       color = col;
   }
   public String getColor() {
       return color;
   }
   public void setSize(int num) {
       size = num;
   }
   public int getSize() {
       return size;
   }
   public void wash() {
       System.out.println("Washing the cloth");
   }
}

class Shirt extends Clothes{
   String type;
   public void setType(String typ) {
       type = typ;
   }
   public String getType() {
       return type;
   }
   public void display() {
       System.out.println(getColor() + " " + type + " of size " + size);
   }
}

class Jeans extends Clothes{
   String material;
```



```
    public void setMaterial(String mat) {
        material = mat;
    }
    public String getMaterial() {
        return material;
    }
    public void display() {
        System.out.println(getColor() + " " + material + " of size " +
size);
    }
}

public class Inherit {
    public static void main(String args[]) {
        Shirt tshirt = new Shirt();
        Jeans denim = new Jeans();

        tshirt.setType("T-shirt");
        tshirt.setColor("Blue");
        tshirt.setSize(32);

        denim.setMaterial("Denim");
        denim.setColor("Blue");
        denim.setSize(40);

        tshirt.wash();
        denim.wash();

        tshirt.display();
        denim.display();
    }
}
```

```
//  ************* Usage of static and non-static  ******************
class Clothes {
    private String color;
    protected int size;
    // static method can be called without any objects
    public static void staticMethod() {
        System.out.println("I am static");
    }
    public void nonStaticMethod() {
        System.out.println("I am non-static");
    }
    public void setColor(String col) {
        color = col;
    }
    public String getColor() {
        return color;
    }
    public void setSize(int num) {
        size = num;
    }
    public int getSize() {
        return size;
    }
    public void wash() {
        System.out.println("Washing the cloth");
    }
}

class Shirt extends Clothes{
    String type;
    public void setType(String typ) {
        type = typ;
    }
    public String getType() {
        return type;
    }
    public void display() {
        System.out.println(getColor() + " " + type + " of size " + size);
    }
 }

 class Jeans extends Clothes{
    String material;
    public void setMaterial(String mat) {
        material = mat;
    }
    public String getMaterial() {
        return material;
    }
    public void display() {
        System.out.println(getColor() + " " + material + " of size " +
 size);
    }
 }

 public class Inherit {
    public static void main(String args[]) {
        Shirt tshirt = new Shirt();
        Jeans denim = new Jeans();

        tshirt.setType("T-shirt");
        tshirt.setColor("Blue");
        tshirt.setSize(32);

        denim.setMaterial("Denim");
        denim.setColor("Blue");
        denim.setSize(40);

        tshirt.wash();
        denim.wash();

        tshirt.display();
        denim.display();
```

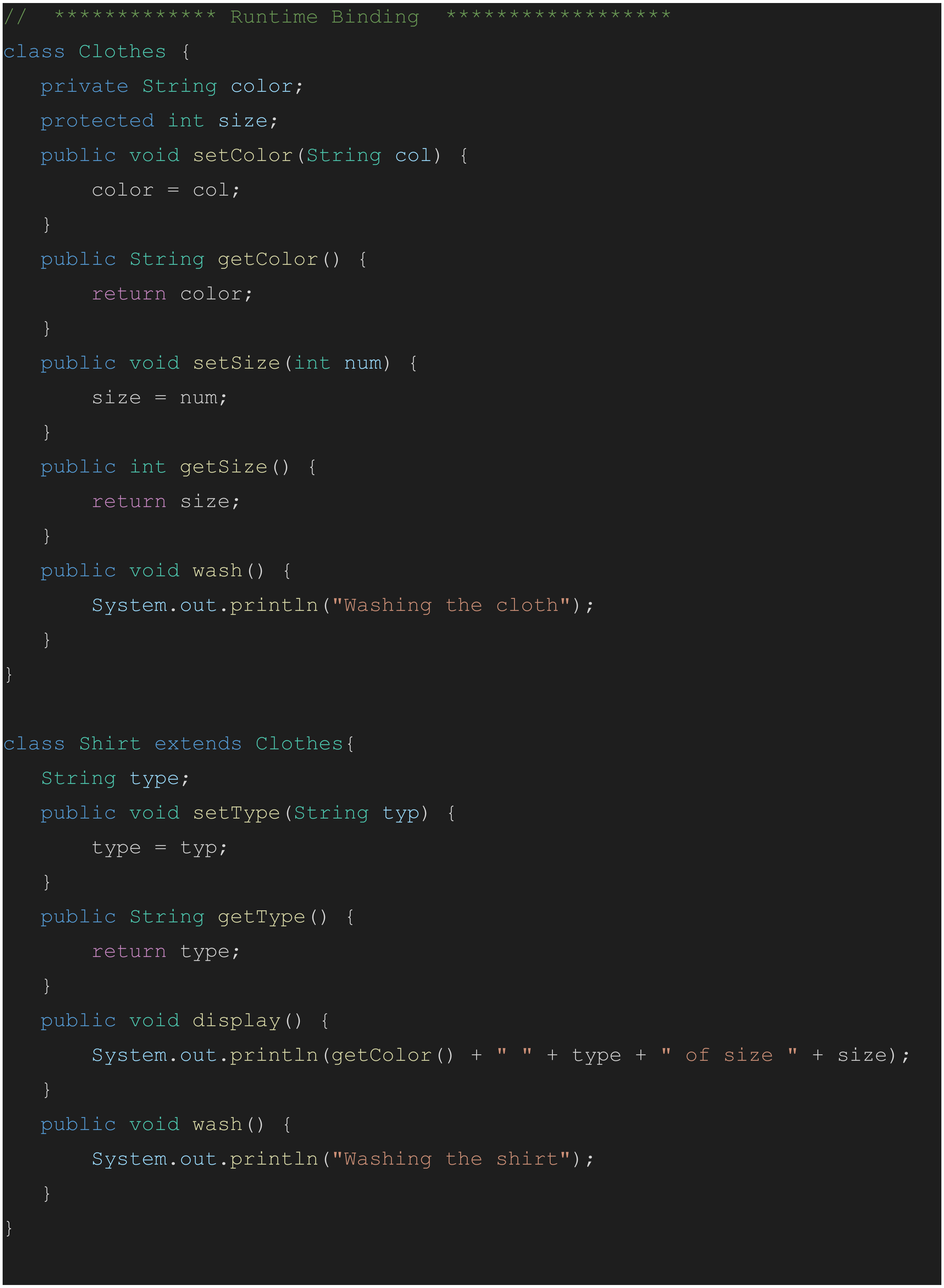
```
        // Non static method cannot be referenced from a static member

        Clothes obj = new Clothes();
        obj.staticMethod();
        obj.nonStaticMethod();

        Clothes.staticMethod();
        //Clothes.nonStaticMethod();
    }
}
```



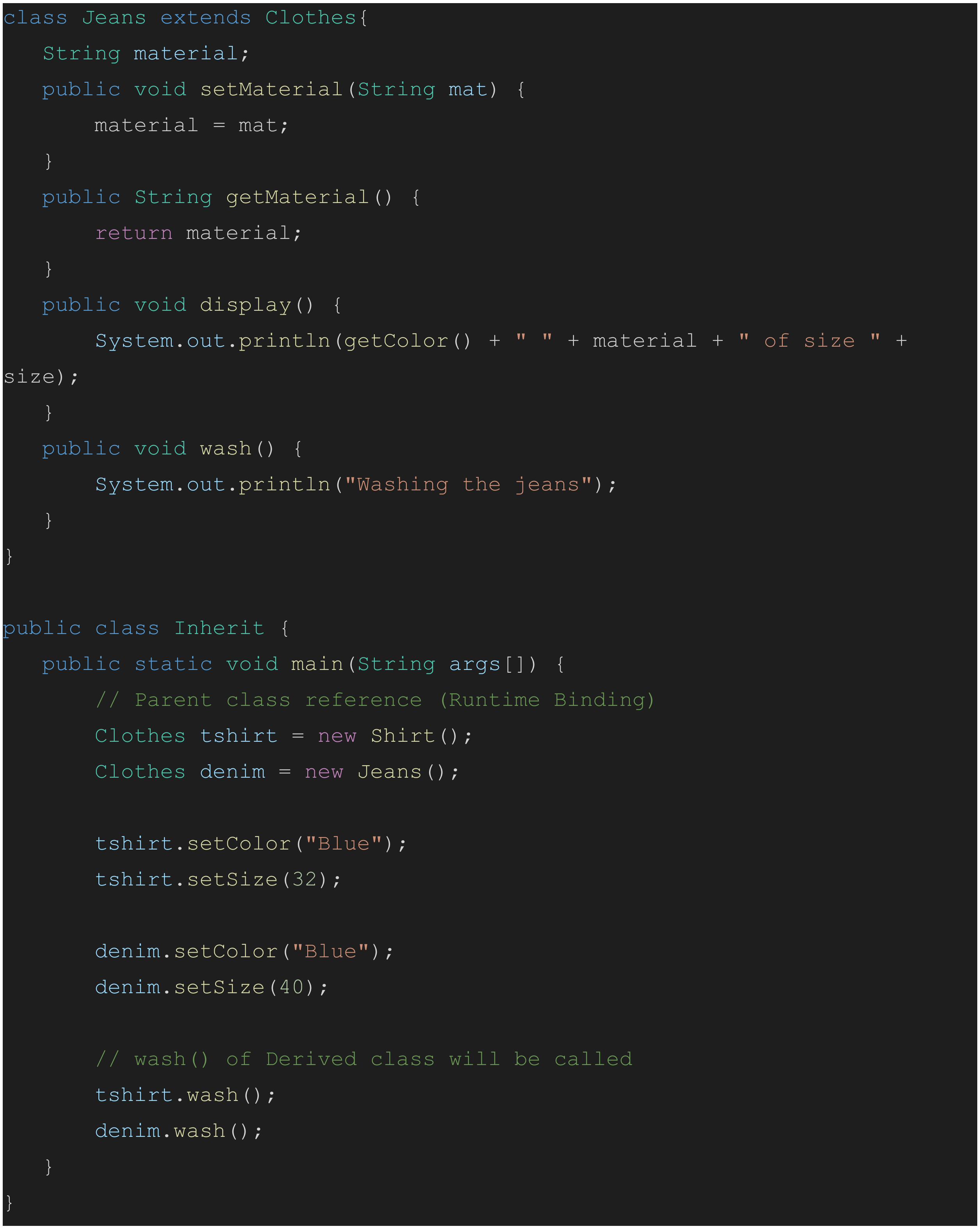
```
//  ************* Overriding Methods  ******************
class Clothes {
   private String color;
   protected int size;
   public void setColor(String col) {
       color = col;
   }
   public String getColor() {
       return color;
   }
   public void setSize(int num) {
       size = num;
   }
   public int getSize() {
       return size;
   }
   public void wash() {
       System.out.println("Washing the cloth");
   }
}

class Shirt extends Clothes{
   String type;
   public void setType(String typ) {
       type = typ;
   }
   public String getType() {
       return type;
   }
   public void display() {
       System.out.println(getColor() + " " + type + " of size " + size);
   }
   // Overriding the wash function
   public void wash() {
       System.out.println("Washing the shirt");
   }
}
```

```
class Jeans extends Clothes{
   String material;
   public void setMaterial(String mat) {
       material = mat;
   }
   public String getMaterial() {
       return material;
   }
   public void display() {
       System.out.println(getColor() + " " + material + " of size " +
size);
   }
   // Overriding the wash function
   public void wash() {
       System.out.println("Washing the jeans");
   }
}

public class Inherit {
   public static void main(String args[]) {
       Shirt tshirt = new Shirt();
       Jeans denim = new Jeans();

       tshirt.setType("T-shirt");
       tshirt.setColor("Blue");
       tshirt.setSize(32);

       denim.setMaterial("Denim");
       denim.setColor("Blue");
       denim.setSize(40);

       // This is called runtime polymorphism
       tshirt.wash();
       denim.wash();

       tshirt.display();
```

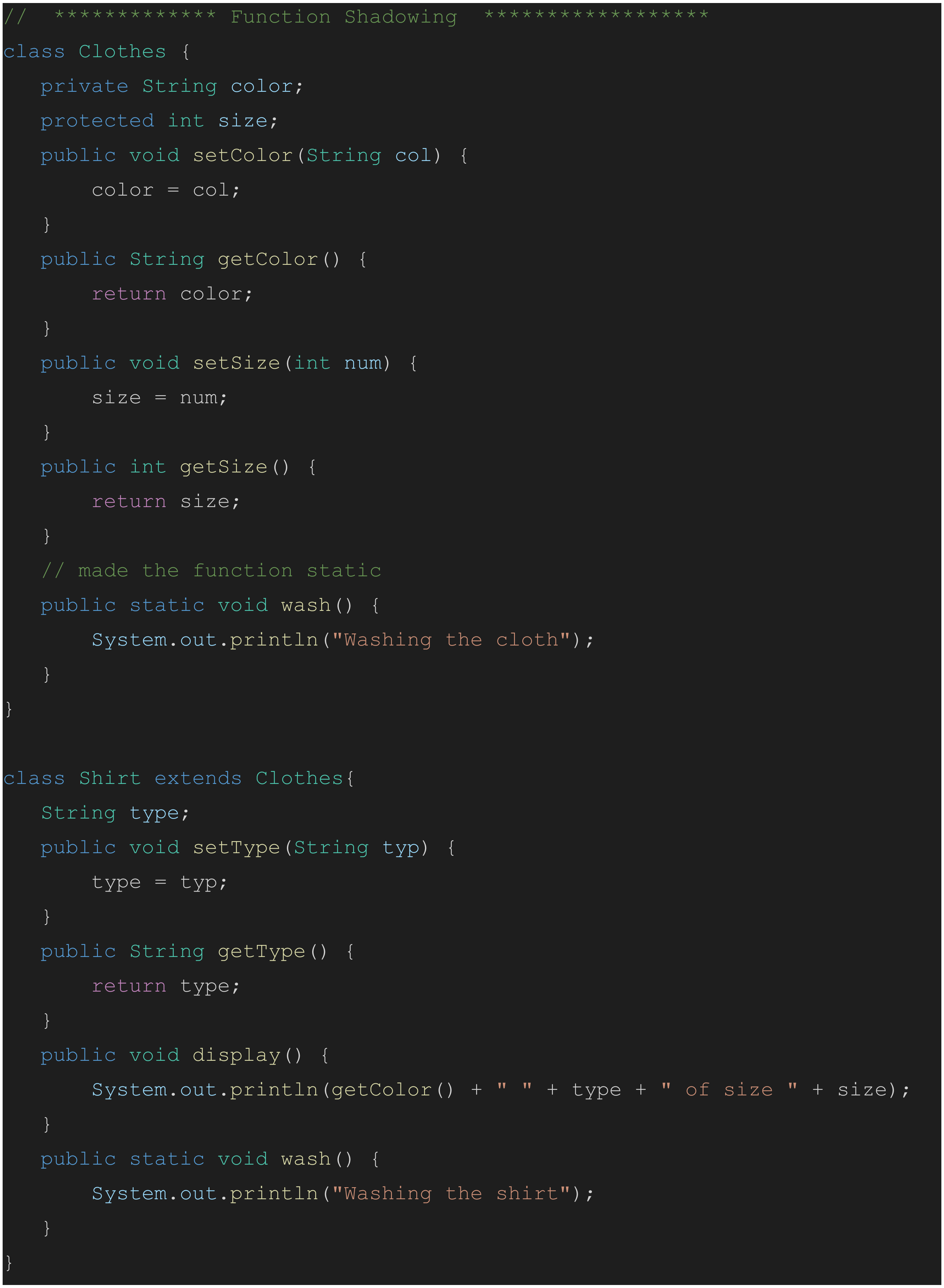
```
        denim.display();
    }
}
```



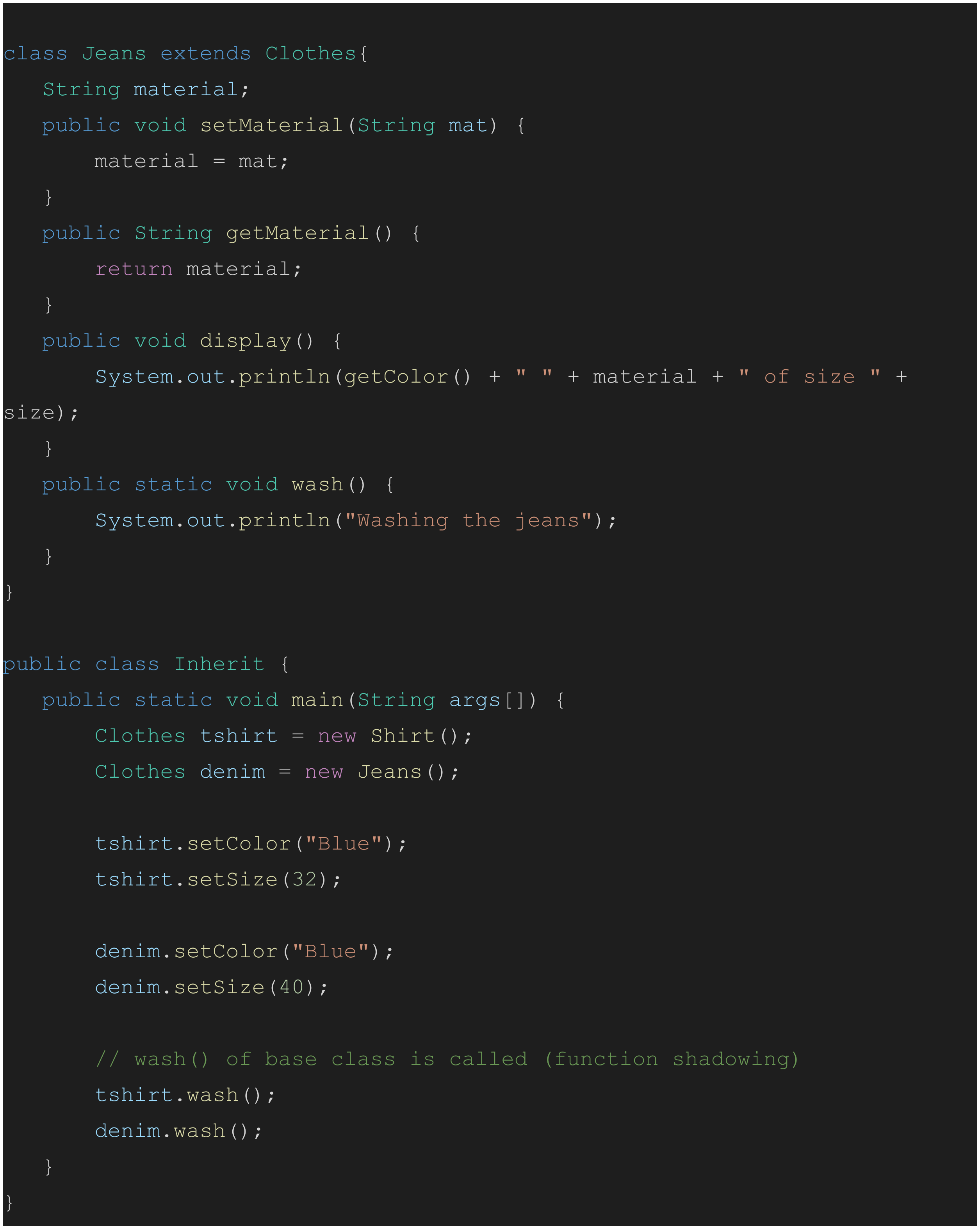
```
//  ************* Runtime Binding  ******************
class Clothes {
   private String color;
   protected int size;
   public void setColor(String col) {
       color = col;
   }
   public String getColor() {
       return color;
   }
   public void setSize(int num) {
       size = num;
   }
   public int getSize() {
       return size;
   }
   public void wash() {
       System.out.println("Washing the cloth");
   }
}

class Shirt extends Clothes{
   String type;
   public void setType(String typ) {
       type = typ;
   }
   public String getType() {
       return type;
   }
   public void display() {
       System.out.println(getColor() + " " + type + " of size " + size);
   }
   public void wash() {
       System.out.println("Washing the shirt");
   }
}
```



```java
class Jeans extends Clothes{
   String material;
   public void setMaterial(String mat) {
       material = mat;
   }
   public String getMaterial() {
       return material;
   }
   public void display() {
       System.out.println(getColor() + " " + material + " of size " +
size);
   }
   public void wash() {
       System.out.println("Washing the jeans");
   }
}

public class Inherit {
   public static void main(String args[]) {
       // Parent class reference (Runtime Binding)
       Clothes tshirt = new Shirt();
       Clothes denim = new Jeans();

       tshirt.setColor("Blue");
       tshirt.setSize(32);

       denim.setColor("Blue");
       denim.setSize(40);

       // wash() of Derived class will be called
       tshirt.wash();
       denim.wash();
   }
}
```



```
//  ************* Function Shadowing  ******************
class Clothes {
   private String color;
   protected int size;
   public void setColor(String col) {
       color = col;
   }
   public String getColor() {
       return color;
   }
   public void setSize(int num) {
       size = num;
   }
   public int getSize() {
       return size;
   }
   // made the function static
   public static void wash() {
       System.out.println("Washing the cloth");
   }
}

class Shirt extends Clothes{
   String type;
   public void setType(String typ) {
       type = typ;
   }
   public String getType() {
       return type;
   }
   public void display() {
       System.out.println(getColor() + " " + type + " of size " + size);
   }
   public static void wash() {
       System.out.println("Washing the shirt");
   }
}
```



```
class Jeans extends Clothes{
    String material;
    public void setMaterial(String mat) {
        material = mat;
    }
    public String getMaterial() {
        return material;
    }
    public void display() {
        System.out.println(getColor() + " " + material + " of size " +
size);
    }
    public static void wash() {
        System.out.println("Washing the jeans");
    }
}

public class Inherit {
    public static void main(String args[]) {
        Clothes tshirt = new Shirt();
        Clothes denim = new Jeans();

        tshirt.setColor("Blue");
        tshirt.setSize(32);

        denim.setColor("Blue");
        denim.setSize(40);

        // wash() of base class is called (function shadowing)
        tshirt.wash();
        denim.wash();
    }
}
```