Lab 3 - Using Methods, Classes and Objects

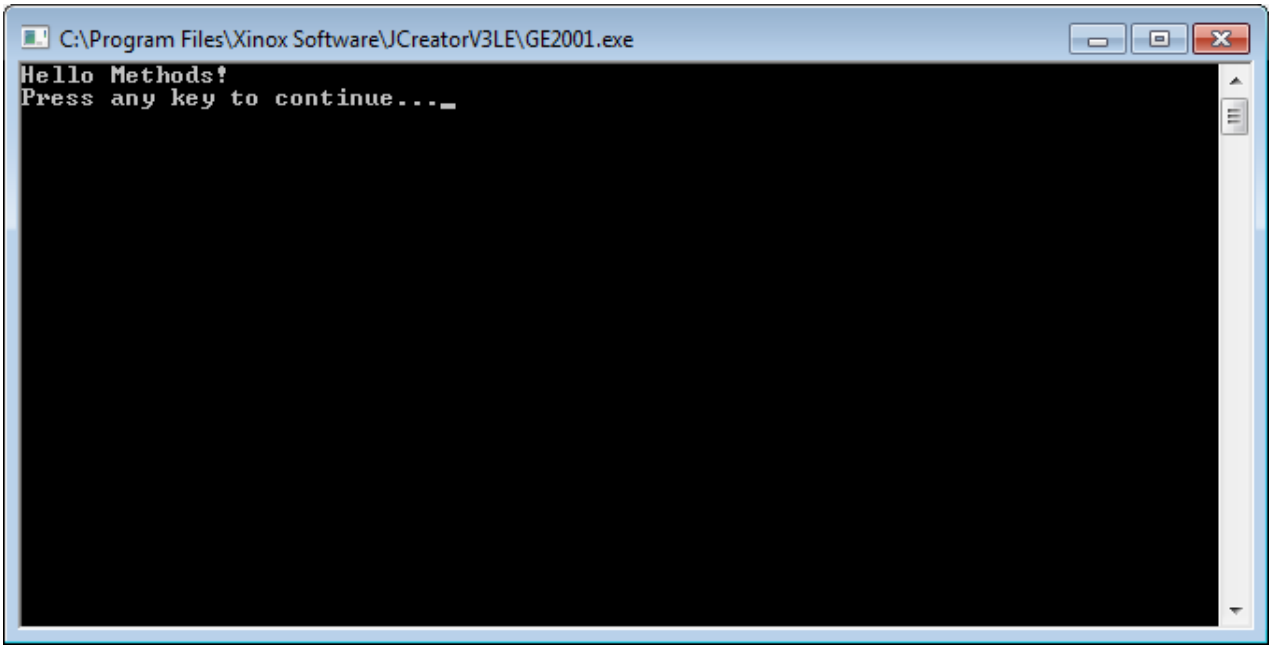
Step 01 – Initialize the IDE

Enter the following code:

Methods.java

```
public class Methods {

   public static void main(String[] args) {

      System.out.println("Hello Methods!"); // Display the string.

   }

}
```

Test the app:



Step 02

Create a Car class with a String attribute called color and a method called start().

Car.java

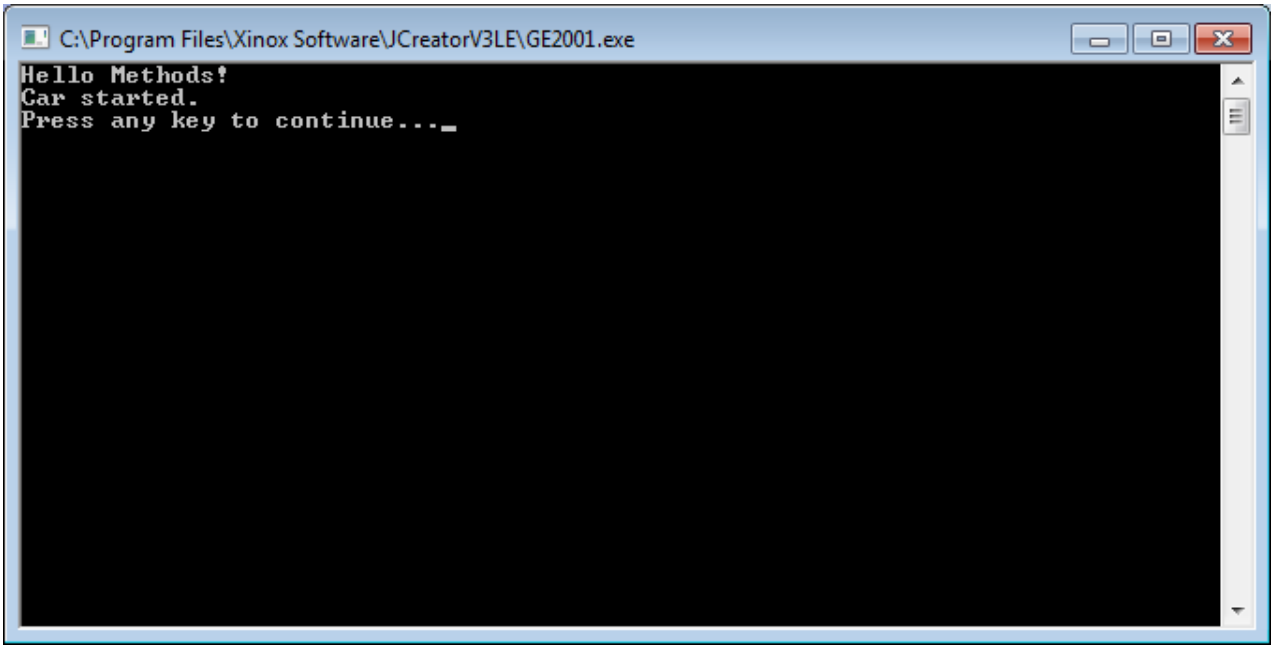
```
public class Car {

	private String color;

	public void start () {

		System.out.println("Car started.");

	}

}
```

Create a Car class and call the start method from the main app:

Methods.java

```
public class Methods {

   public static void main(String[] args) {

      System.out.println("Hello Methods!");

      Car mustang = new Car();

      mustang.start();

   }

}
```

Test the app:


C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Hello Methods!
Car started.
Press any key to continue...

Step 03 – Add a Constructor

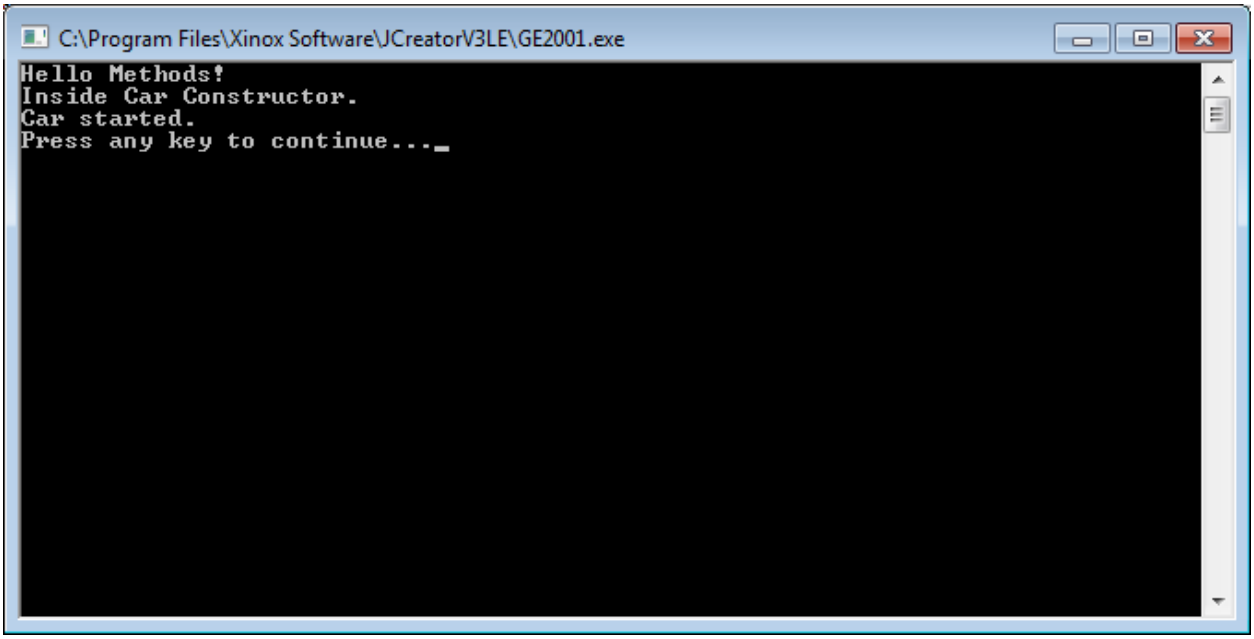
Car.java

```
public class Car {

	private String color;

	public Car() {

		System.out.println("Inside Car Constructor.");
	}

	public void start () {

		System.out.println("Car started.");

	}

}
```

Methods.java

```
public class Methods {

   public static void main(String[] args) {

      System.out.println("Hello Methods!");

      Car mustang = new Car();

      mustang.start();

   }

}
```

Test the app:


C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Hello Methods!
Inside Car Constructor.
Car started.
Press any key to continue...

Step 04 – Add Accessor Methods for the *color* Attribute

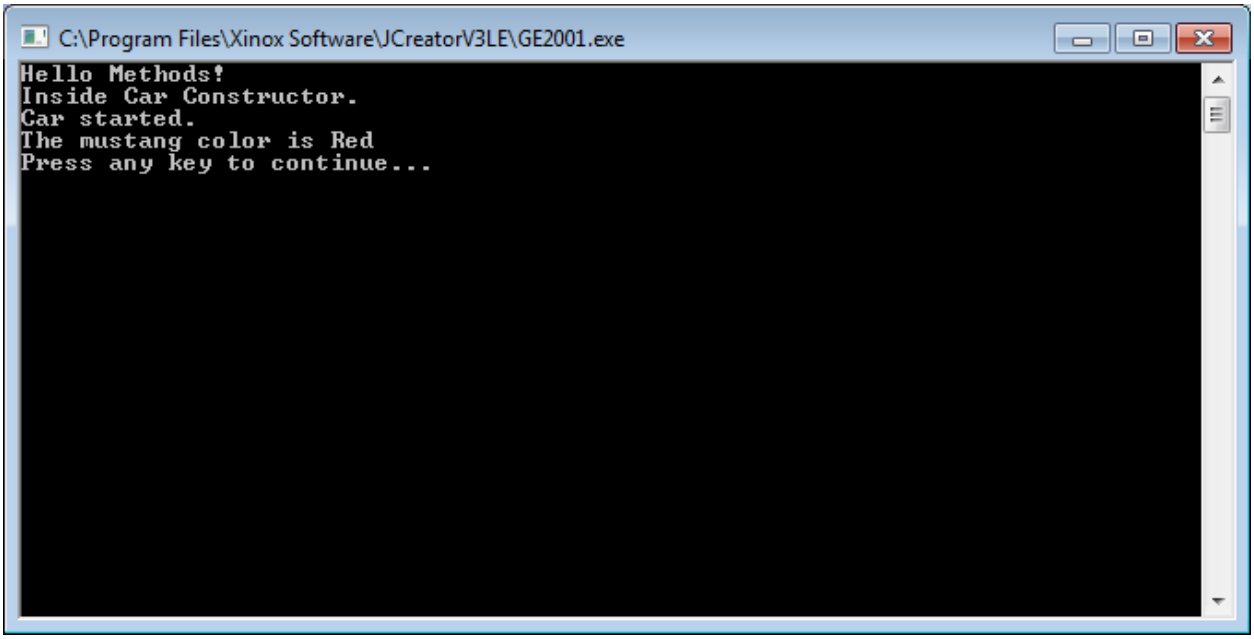
Car.java

```
public class Car {

	private String color;

	public Car() {

		System.out.println("Inside Car Constructor.");
	}

	public void setColor (String c) {

		color = c;

	}

	public String getColor () {

		return color;

	}
	public void start () {

		System.out.println("Car started.");

	}

}
```

Methods.java

```
public class Methods {

   public static void main(String[] args) {

      System.out.println("Hello Methods!");

      Car mustang = new Car();

      mustang.start();

      mustang.setColor("Red");

      System.out.println("The mustang color is " + mustang.getColor());

   }

}
```

Test the app:


C:\Program Files\Xinox Software\JCreatorV3LE\GE2001.exe
Hello Methods!
Inside Car Constructor.
Car started.
The mustang color is Red
Press any key to continue...

Step 05 – Add a parent Class Called Vehicle and have Car Inherit from It

Vehicle.java

```
public class Vehicle {

	public Vehicle() {

		System.out.println("Inside Vehicle Constructor.");
	}

}
```

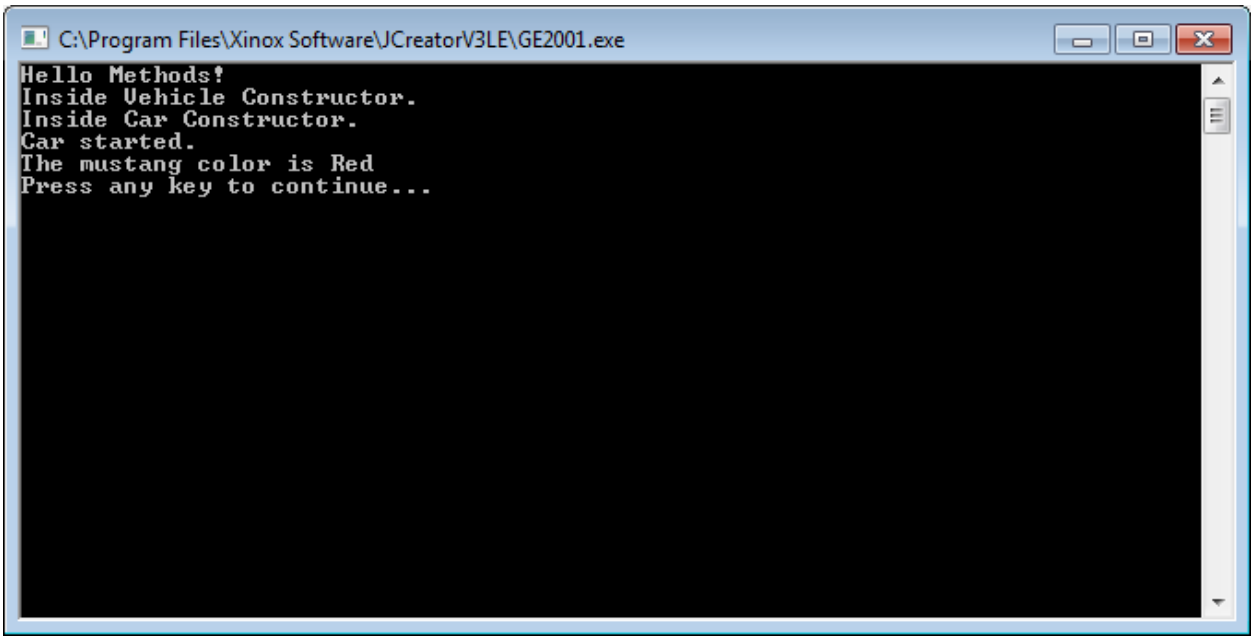
Car.java

```
public class Car extends Vehicle{

	private String color;

	public Car() {

		System.out.println("Inside Car Constructor.");
	}

	public void setColor (String c) {

		color = c;

	}

	public String getColor () {

		return color;

	}
	public void start () {

		System.out.println("Car started.");

	}

}
```

Methods.java

```
public class Methods {

   public static void main(String[] args) {

      System.out.println("Hello Methods!");

      Car mustang = new Car();

      mustang.start();

      mustang.setColor("Red");

      System.out.println("The mustang color is " + mustang.getColor());

   }

}
```

Test the app:



Step 06 – Create an Overloaded Method in Car

Vehicle.java

```
public class Vehicle {

	public Vehicle() {

		System.out.println("Inside Vehicle Constructor.");
	}

}
```

Car.java

```
public class Car extends Vehicle{

	private String color;

	public Car() {

		System.out.println("Inside Car Constructor.");
	}

	public Car(String c) {

		System.out.println("Inside Overloaded Car Constructor.");

      color = c;
	}


	public void setColor (String c) {

		color = c;

	}

	public String getColor () {

		return color;

	}
	public void start () {

		System.out.println("Car started.");

	}

}
```

Methods.java

```
public class Methods {

   public static void main(String[] args) {

      System.out.println("Hello Methods!");

      Car mustang = new Car("Blue");

      System.out.println("The mustang color after the contructor is " + mustang.getColor());

            mustang.start();

            mustang.setColor("Red");

      System.out.println("The mustang color after the setter is " + mustang.getColor());

   }

}
```

Test the app: