1. Log in
2. Unit 4
   - Writing Methods
   - Recursion
   - Classes, Methods, and Objects

Mar 9-11:28 AM

Reminder - Constructing an Object

```
Car my1stCar = new Car(an integer, a String);
```

- Car: type of object (a Car)
- my1stCar: name of the object (Car name)
- new Car: required to make a new Car object
- (an integer, a String): parameters = stuff needed to build the Car

May 18-1:29 PM

Previous Lesson: We had written 2 different classes ...

Car Constructor Class

```
package Lesson15;

/**
 * @author Matthew_Henderson
 */
public class Car
{
        public int year;
        public String type;

        public Car(int carYear, String carType)
        {
                year = carYear;
                type = carType;
        }
}
```

elmoYouCanDriveMyCar Class
The Main program!

```
package Lesson15;

/**
 * @author Matthew_Henderson
 */
public class elmoYouCanDriveMyCar {
   public static void main(String[] args) {
      Car myFirstCar = new Car(1989, "Escort");
      Car myCurrentCar = new Car(2015, "Porsche");
   }
}
```

The main program will use the Car constructor class when needing to build a Car object.

Apr 21-1:43 PM

New Concept - Encapsulation

*** We don't need 2 different classes ***

```
package unit4;
public class Car
{
   public int year;
   public String type;

   public Car(int carYear, String carType) {
      year = carYear;
      type = carType;
   }
   public static void main(String[] args) {
      Car myFirstCar = new Car(1989, "Escort");
      Car myCurrentCar = new Car(2015, "Porsche");
      System.out.println(myFirstCar.year);
   }
}
```

Apr 21-1:43 PM

In fact, we can encapsulate the entire program into one class ...

```
package Lesson15;
public class Car
{
        public int year;
        public String type;
        public boolean running;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
        }
        public void startCar( ) {
                running = true;
        }
        public void turnCarOff( ) {
                running = false;
        }
        public static void main(String[] args) {
            Car myFirstCar = new Car(1989, "Escort");
            Car myCurrentCar = new Car(2015, "Porsche");
        }
}
```

Types of Methods

Constructor Method
(builds an object when called)

*** Every time it is called ***
*** it creates an instance ***

Apr 21-1:43 PM

In fact, we can encapsulate the entire program into one class ...

```
package Lesson15;
public class Car
{
        public int year;
        public String type;
        public boolean running;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
        }
        public void startCar( ) {
                running = true;
        }
        public void turnCarOff( ) {
                running = false;
        }
        public static void main(String[] args) {
            Car myFirstCar = new Car(1989, "Escort");
            Car myCurrentCar = new Car(2015, "Porsche");
        }
}
```

Types of Methods

Constructor Method
(builds an object when called)

Mutator Method
(changes, or mutates, something about an object)

Apr 21-1:43 PM

In fact, we can encapsulate the entire program into one class ...

```
package Lesson15;

public class Car
{
        public int year;
        public String type;
        public boolean running;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
        }

        public void startCar( ) {
                running = true;
        }

        public void turnCarOff( ) {
                running = false;
        }
        public static void main(String[] args) {
            Car myFirstCar = new Car(1989, "Escort");
            Car myCurrentCar = new Car(2015, "Porsche");
        }
}
```

Types of Methods

Constructor Method
(builds an object when called)

Mutator Method
(changes, or mutates, something about an object)

Mutator Method
(mutates/changes running state of a car)

Apr 21-1:43 PM

In fact, we can encapsulate the entire program into one class ...

```
package Lesson15;

public class Car
{
        public int year;
        public String type;
        public boolean running;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
        }

        public void startCar( ) {
                running = true;
        }

        public void turnCarOff( ) {
                running = false;
        }
        public static void main(String[] args) {
            Car myFirstCar = new Car(1989, "Escort");
            Car myCurrentCar = new Car(2015, "Porsche");
        }
}
```

Types of Methods

Constructor Method
(builds an object when called)

Mutator Method
(changes, or mutates, something about an object)

Mutator Method
(mutates/changes running state of a car)

Main Method
The list of directions to follow

Apr 21-1:43 PM

So now what? Do you want the car to move???

Here are some things to consider ...

1. We might need to track the car's speed (speedometer).
2. We might need to speed up the car (accelerate).
3. We might need to slow down the car (brake).
4. We might need to read the speedometer!

*** We will analyze these things ***

May 19-10:59 AM

Here is our current Car class ...

Here's what we are considering ...

```
package Lesson15;

public class Car
{
        public int year;
        public String type;
        public boolean running;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
        }

        public void startCar( ) {
                running = true;
        }

        public void turnCarOff( ) {
                running = false;
        }
}
```

Doing after the car is started is logical

We will need to track the speedometer (an integer variable)

We will need perform an action on the speedometer... a method that increases it! (accelerate)

We will need to perform another action that decreases speedometer! (brake)

We will need to be able to get our current speed (read speedometer)

May 19-10:59 AM

In order to read the speedometer ...

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0;
        }

        public void startCar( ) {
                running = true;
        }

        public void turnCarOff( ) {
                running = false;
        }

        public int getSpeed( ) {
                return currentSpeed;
        }
}
```

We need an integer to store the speed!

When a car is made, it's speed is set to zero!

We can create a method that reads our current speed... call it getSpeed( ) (see next slide for getSpeed( ) analysis)

May 19-10:59 AM

Here's the method ... it is simple!

```
public int getSpeed( )
{
        return currentSpeed;
}
```

This method simply reads the car's speed and returns it to the program.

Public - Remember it simply means this method is open for all to use! We want to access it!

int - For now, it states that this method will be returning an integer answer!

getSpeed( ) - Simply the name of our method (no parameters needed!). We made our own method!

May 19-10:59 AM

Now in order to accelerate, we need another method (action)!

```
public void accelerate( )
{
        currentSpeed+=5;
}
```

This method simply adds 5 to the current speed of the car.

Public - Remember it simply means this method is open for all to use!

void - Remember, void simply alters things ... does not return anything to the program. That is exactly what we want to do ... change the speed but not tell us anything.

accelerate( ) - Simply the name of our method (no parameters needed!). We made our own method!

May 19-10:59 AM

Analyzing our new class with the new methods ...

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }
}
```

Want the current speed of a car? Call this method ... it acts like a radar gun!

A new method that increases the currentSpeed, called accelerate( )

Can you make the brake method??? (think before going to next slide!)

May 19-10:59 AM

Now in order to brake, we need another method (action)!

```
public void brake( )
{
        currentSpeed-=8;
        if(currentSpeed < 0)
                currentSpeed = 0;
}
```

This method says we can brake faster than accelerating. And if speed goes negative, we are stopped!

Public - Remember it simply means this method is open for all to use!

void - Remember, void simply alters things ... does not return anything to the program.

brake( ) - Simply the name of our method (no parameters needed!). We made our own method!

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                        currentSpeed=0;
        }
}
```

Here's our final program (to this point) ...

Types of Methods

Constructor Method
(builds an object when called)

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                        currentSpeed=0;
        }
}
```

Here's our final program (to this point) ...

Types of Methods

Constructor Method
(builds an object when called)

Mutator Method
(changes running state of a car object)

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                        currentSpeed=0;
        }
}
```

Here's our final program (to this point) ...

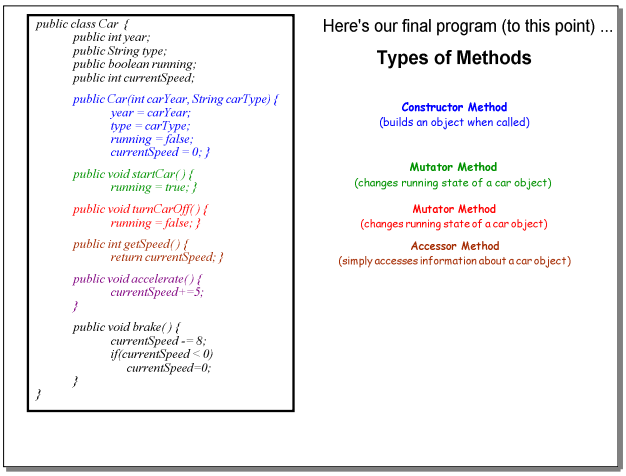
Types of Methods

Constructor Method
(builds an object when called)

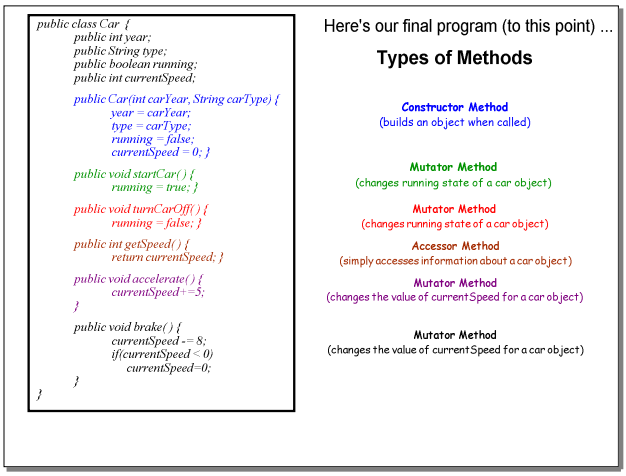
Mutator Method
(changes running state of a car object)

Mutator Method
(changes running state of a car object)

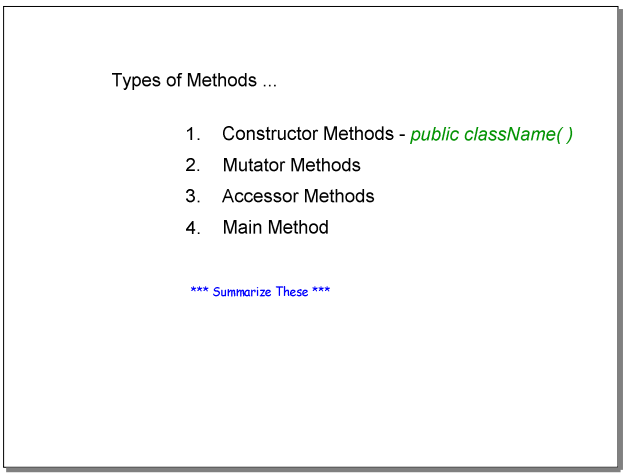
May 19-10:59 AM



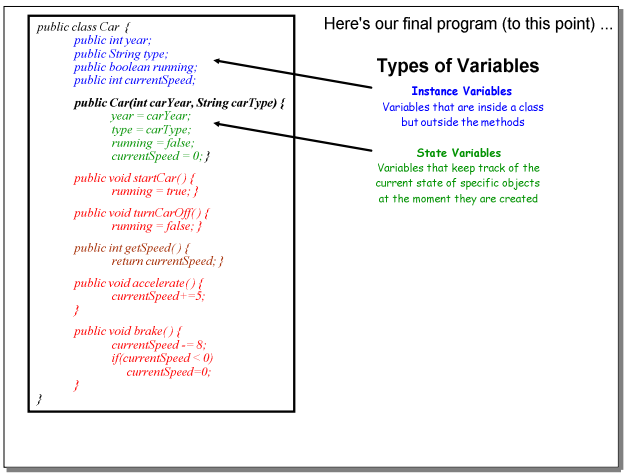
May 19-10:59 AM



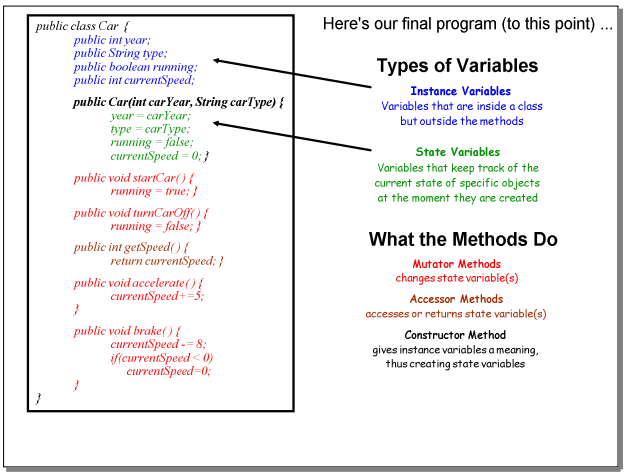
May 19-10:59 AM



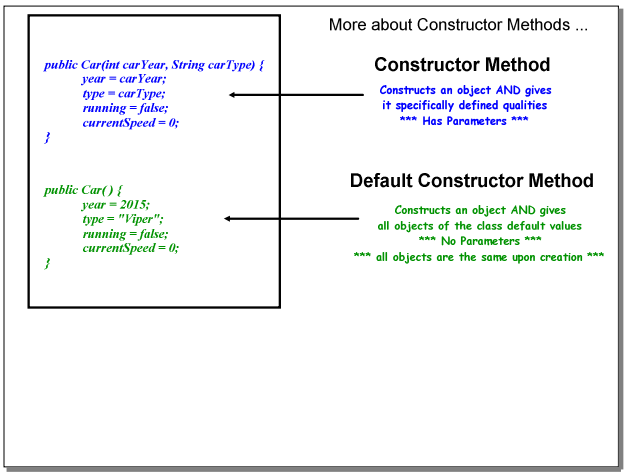
Nov 6-12:44 PM



May 19-10:59 AM



May 19-10:59 AM



May 19-10:59 AM

Now it is time to add a main method and perform some action to the car(s)

Nov 6-2:10 PM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                    currentSpeed=0;
        }
}
```

Analyze what the following does ...

```
public static void main(String[] args) {

    Car car2 = new Car(2014, "Corolla");

    System.out.println("The "+car2.type+" is running: "+car2.running);

}
```

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                    currentSpeed=0;
        }
}
```

Analyze what the following does ...

```
public static void main(String[] args) {

    Car car2 = new Car(2014, "Corolla");

    System.out.println("The "+car2.type+" is running: "+car2.running);

    car2.startCar();

    System.out.println("The "+car2.type+" is running: "+car2.running);

}
```

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                    currentSpeed=0;
        }
}
```

Analyze what the following does ...

```
public static void main(String[] args) {

    Car car2 = new Car(2014, "Corolla");

    System.out.println("The "+car2.type+" is running: "+car2.running);

    car2.startCar();

    System.out.println("The "+car2.type+" is running: "+car2.running);

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

    for(int i=1 ; i<=3 ; i++)
        car2.accelerate();

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

}
```

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                    currentSpeed=0;
        }
}
```

Analyze what the following does ...

```
public static void main(String[] args) {

    Car car2 = new Car(2014, "Corolla");

    System.out.println("The "+car2.type+" is running: "+car2.running);

    car2.startCar();

    System.out.println("The "+car2.type+" is running: "+car2.running);

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

    for(int i=1 ; i<=3 ; i++)
        car2.accelerate();

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

    car2.brake();

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

}
```

May 19-10:59 AM

```
public class Car {
        public int year;
        public String type;
        public boolean running;
        public int currentSpeed;

        public Car(int carYear, String carType) {
                year = carYear;
                type = carType;
                running = false;
                currentSpeed = 0; }

        public void startCar( ) {
                running = true; }

        public void turnCarOff( ) {
                running = false; }

        public int getSpeed( ) {
                return currentSpeed; }

        public void accelerate( ) {
                currentSpeed+=5;
        }

        public void brake( ) {
                currentSpeed -= 8;
                if(currentSpeed < 0)
                    currentSpeed=0;
        }
}
```

Analyze what the following does ...

```
public static void main(String[] args) {

    Car car2 = new Car(2014, "Corolla");

    System.out.println("The "+car2.type+" is running: "+car2.running);

    car2.startCar();

    System.out.println("The "+car2.type+" is running: "+car2.running);

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

    for(int i=1 ; i<=3 ; i++)
        car2.accelerate();

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

    car2.brake();

    System.out.println(car2.type+" current speed is "+car2.getSpeed());

    car2.turnCarOff();

    System.out.println("Running is "+car2.running+" but " +
                       "speed is " +car2.getSpeed());
}
```

May 19-10:59 AM

Things to do ...

1. Do you have Unit 4 WS01-04 completed?
2. Be wrapping up Unit 04 WS05 - Classes & Objects 1
3. Work on Unit 04 WS06 - Methods and Objects 2

Nov 6-3:25 PM