# Classes, Objects, Methods


(A classy 1/6 scale working model  by Louis Chenot)

- A class is a template
  - It defines the form of an object
  - It specifies data held in the object
  - It specifies methods that operate on the data

- A class is not the object itself.

- A physical representation of the class does not exist until an object of that class is created

# General Form of a Class

```
class classname {
  // declare instance variables
  type var1;
  type var2;
  // ...
  type varN;

  // declare methods
  type method1(parameters) {
     // body of method
  }
  type method2(parameters) {
     // body of method
  }
  // ...
  type methodN(parameters) {
     // body of method
  }
}
```

# Defining a Class

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon
}
```

A class with just data in it

# Defining a Class

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon
}
```

## Accessing members of a class

Use the *dot operator* (.)

*object.member*

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon
}

 class DemoOneVehicle {
   public static void main(String args[]) {
     Vehicle minivan = new Vehicle();

        // assign values to fields in minivan
     minivan.passengers = 7;
     minivan.fuelcap = 16;
     minivan.mpg = 21;

     System.out.print("Minivan MPG = " + minivan.mpg );
   }
 }
```

```
class DemoOneVehicle {
  public static void main(String args[]) {
     Vehicle minivan = new Vehicle();

     int range;

        // assign values to fields in minivan
     minivan.passengers = 7;
     minivan.fuelcap = 16;
     minivan.mpg = 21;

     range = minivan.fuelcap * minivan.mpg;

     System.out.println("Minivan range =" + range );
  }
}
```

doing math on members from a class

```
class DemoTwoVehicles {
  public static void main(String args[]) {
    Vehicle minivan = new Vehicle();
    Vehicle sportscar = new Vehicle();

      // assign values to fields in minivan
    minivan.passengers = 7;
    minivan.fuelcap = 16;
    minivan.mpg = 21;

     // assign values to fields in sportscar
    sportscar.passengers = 2;
    sportscar.fuelcap = 14;
    sportscar.mpg = 12;

    System.out.print("Minivan MPG = "+ minivan.mpg );
    System.out.print("Sportscar MPG = "+ sportscar.mpg);
  }
}
```

# Defining a Class - data & method

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon

                      // and now define a method
    void range( ) {
        System.out.println("Range is " + fuelcap * mpg);
    }
}
```

A class with data and a method

```
class DemoOneVehicle {
  public static void main(String args[]) {
    Vehicle minivan = new Vehicle();

          // assign values to fields in minivan
    minivan.passengers = 7;
    minivan.fuelcap = 16;
    minivan.mpg = 21;

    minivan.range( );

  }
}
```

“call” a method to print out the range

# Defining a Class - data & method

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon

                      // and now define a method
    int range( ) {
        return fuelcap * mpg;
    }
}
```

A class with data and a method that returns a value

```
class DemoOneVehicle {
  public static void main(String args[]) {
     Vehicle minivan = new Vehicle( );
     int myRange;

          // assign values to fields in minivan
     minivan.passengers = 7;
     minivan.fuelcap = 16;
     minivan.mpg = 21;

     myRange = minivan.range( );
     System.out.println("My range is " + myRange;
  }
}
```

"call" a method to get a value and then display it in the main

# Defining a Class containing data & method

## where method accepts a parameter

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon

                      // and now define a method
    int range(int speedfactor ) {
        if (speedfactor == 0)
            return fuelcap * mpg;
        else
            return (int) (fuelcap * mpg * 0.8);
    }
}
```

This method requires a parameter used to calculate the value it returns. Valid parameter values are 0 or 1 where 1 means reduce range to 80% due to high speed driving.

# Now create an object and use its method

```
class DemoOneVehicle {
  public static void main(String args[]) {
    Vehicle minivan = new Vehicle();
    int myRange;

        // assign values to fields in minivan
    minivan.passengers = 7;
    minivan.fuelcap = 16;
    minivan.mpg = 21;

    myRange = minivan.range(1);
    System.out.println("My range is " + myRange);
  }
}
```

"call" the Vehicle method to get a value using a parameter and then display it in the main
The parameter 1 means derate the range due to high speed driving.

# Adding a constructor to a class

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon

        // this is a constructor for Vehicle
    Vehicle( ) {
        passengers = 7;
        fuelcap = 16;
        mpg = 21;
    }
}
```

Initializing data in a class using a constructor. In this case, fixed initial values. The constructor is called by new when a Vehicle object is created.

# Adding a constructor to a class

## With initial values passed to it

```
class Vehicle {
    int passengers;   // number of passengers
    int fuelcap;      // fuel capacity in gallons
    int mpg;          // fuel consumption in miles per gallon

        // this is a constructor with parameters for Vehicle
    Vehicle(int p, int f, int m ) {
        passengers = p;
        fuelcap = f;
        mpg = m;
    }
}
```

Initializing data in a class using a constructor with parameters passed to it.
The constructor is called by new when a Vehicle object is created.

# Create an object and pass initial values

```
class DemoOneVehicle {
  public static void main(String args[]) {
    Vehicle minivan = new Vehicle(7,16,21);

    int myRange;

    myRange = minivan.range(1);
    System.out.println("My range is " + myRange);
  }
}
```

Values for the object's data are passed as parameters when the object is created.