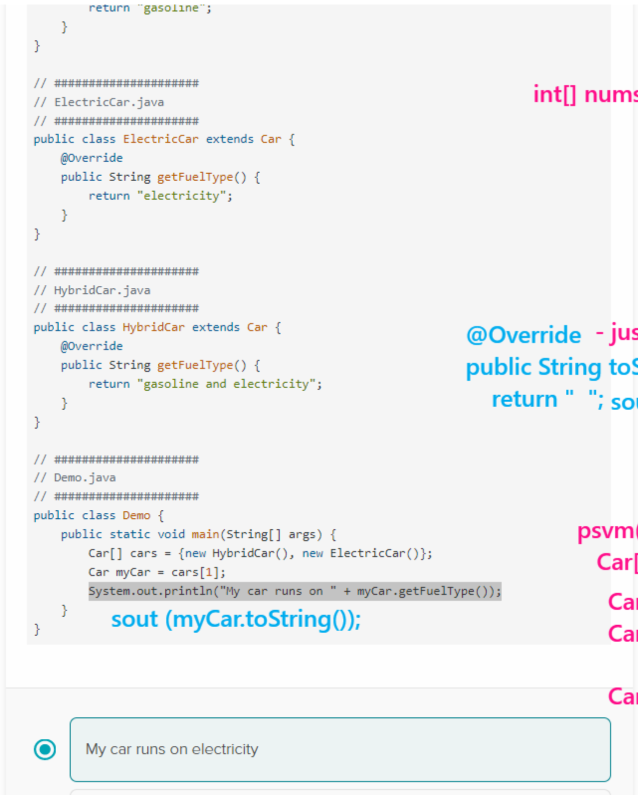


```
        return "gasoline";
    }
}

// #####################
// ElectricCar.java
// #####################
public class ElectricCar extends Car {
    @Override
    public String getFuelType() {
        return "electricity";
    }
}

// #####################
// HybridCar.java
// #####################
public class HybridCar extends Car {
    @Override
    public String getFuelType() {
        return "gasoline and electricity";
    }
}

// #####################
// Demo.java
// #####################
public class Demo {
    public static void main(String[] args) {
        Car[] cars = {new HybridCar(), new ElectricCar()};
        Car myCar = cars[1];
        System.out.println("My car runs on " + myCar.getFuelType());
    }
}
```

sout (myCar.toString());

[✓ My car runs on electricity]

int[] nums = {2, 3, 4}

@Override - just communicates

public String toString() {
    return " "; sout(" ");

psvm(S[] a) {

anonymous

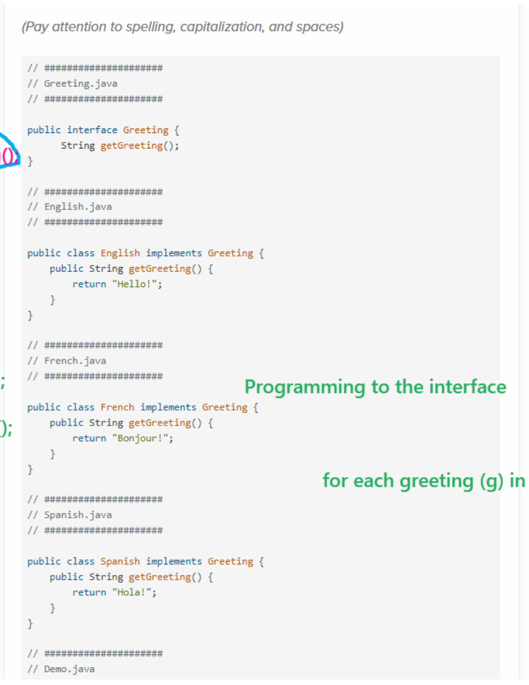
{new HybridCar(), new ElectricCar() }

Car[] cars = new Car[2];

Car car1 = new HybridCar();

Car car2 = new ElectricCar();

Car[] cars = {car1, car2}



(Pay attention to spelling, capitalization, and spaces)

```
// #####################
// Greeting.java
// #####################

public interface Greeting {
      String getGreeting();
}

// #####################
// English.java
// #####################

public class English implements Greeting {
    public String getGreeting() {
        return "Hello!";
    }
}

// #####################
// French.java
// #####################

public class French implements Greeting {
    public String getGreeting() {
        return "Bonjour!";
    }
}

// #####################
// Spanish.java
// #####################

public class Spanish implements Greeting {
    public String getGreeting() {
        return "Hola!";
    }
}

// #####################
// Demo.java
```

Greeting greeting = new French();

~~Greeting greeting = new Greeting();~~

~~greeting.getGreeting(); ????~~

interface   class

List<String> myList = new ArrayList<>();

Map<K, V> myMap = new HashMap<>();

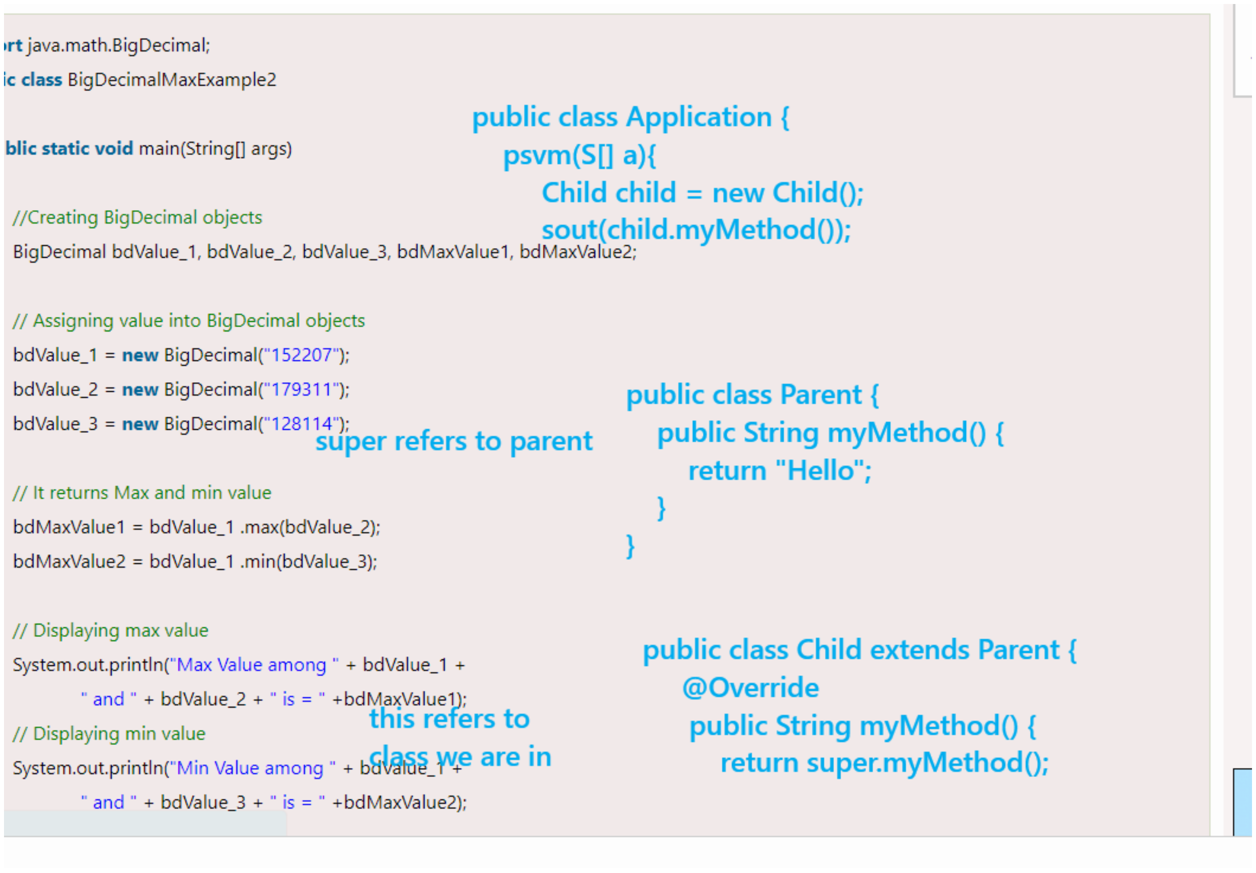
Programming to the interface

for each greeting (g) in array called greetings

Greeting[]
greetings

g

new French()
English()
Spanish



```
rt java.math.BigDecimal;
c class BigDecimalMaxExample2

blic static void main(String[] args)

 //Creating BigDecimal objects
 BigDecimal bdValue_1, bdValue_2, bdValue_3, bdMaxValue1, bdMaxValue2;

 // Assigning value into BigDecimal objects
 bdValue_1 = new BigDecimal("152207");
 bdValue_2 = new BigDecimal("179311");
 bdValue_3 = new BigDecimal("128114");

 // It returns Max and min value
 bdMaxValue1 = bdValue_1 .max(bdValue_2);
 bdMaxValue2 = bdValue_1 .min(bdValue_3);

 // Displaying max value
 System.out.println("Max Value among " + bdValue_1 +
          " and " + bdValue_2 + " is = " +bdMaxValue1);
 // Displaying min value
 System.out.println("Min Value among " + bdValue_1 +
          " and " + bdValue_3 + " is = " +bdMaxValue2);
```

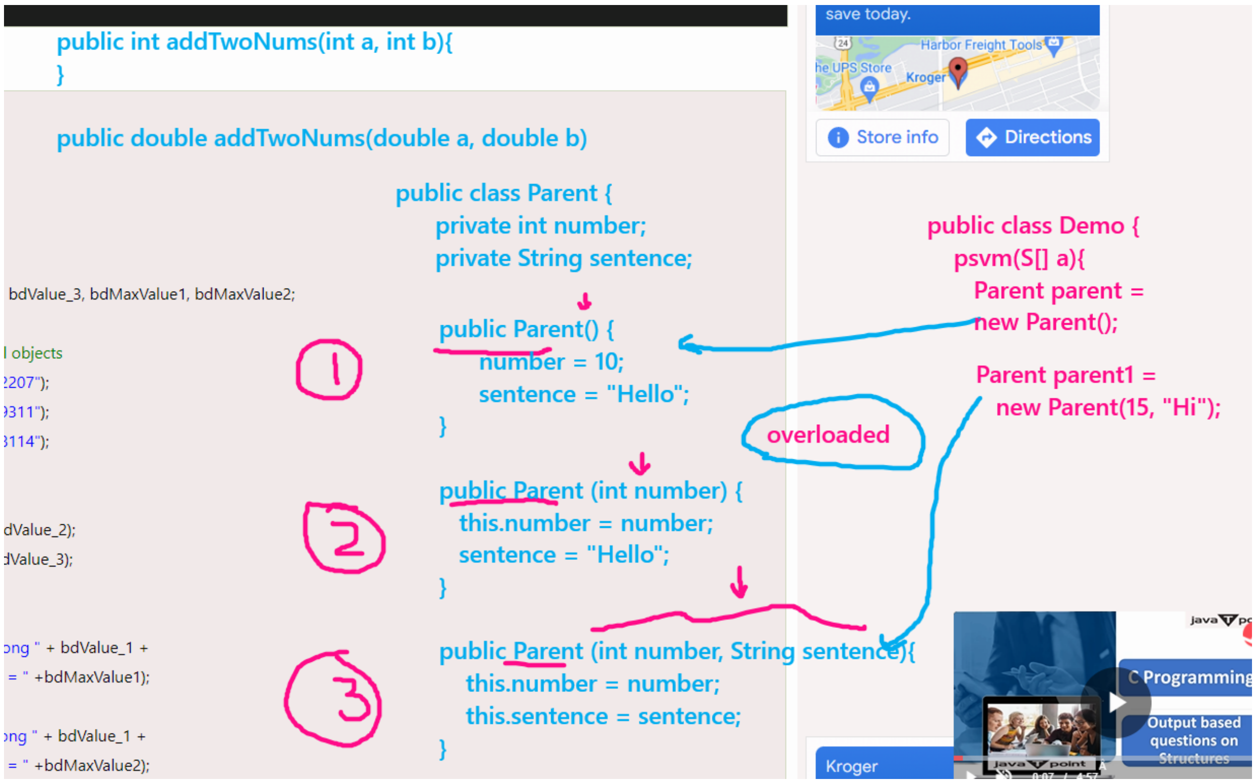
```
public class Application {
  psvm(S[] a){
     Child child = new Child();
     sout(child.myMethod());
```

super refers to parent

```
public class Parent {
  public String myMethod() {
     return "Hello";
  }
}
```

```
public class Child extends Parent {
   @Override
   public String myMethod() {
      return super.myMethod();
```

this refers to class we are in



```
public int addTwoNums(int a, int b){
}

public double addTwoNums(double a, double b)
```

```
public class Parent {
   private int number;
   private String sentence;

 ① public Parent() {
      number = 10;
      sentence = "Hello";
   }

 ② public Parent (int number) {
     this.number = number;
     sentence = "Hello";
   }

 ③ public Parent (int number, String sentence){
      this.number = number;
      this.sentence = sentence;
   }
```

overloaded

```
public class Demo {
  psvm(S[] a){
    Parent parent =
    new Parent();

    Parent parent1 =
      new Parent(15, "Hi");
```

```
 bdValue_3, bdMaxValue1, bdMaxValue2;
l objects
207");
311");
114");
dValue_2);
dValue_3);
ong " + bdValue_1 +
= " +bdMaxValue1);
ong " + bdValue_1 +
= " +bdMaxValue2);
```