extends -- explicitly telling Java what class you are a child of

no keyword extends -- Object is the parent

SELECT ALL CORRECT OPTIONS

- [ ] String
- [x] Object
- [x] Shape
- [ ] Square
- [ ] Triangle
- [ ] Ellipse

✓ Correct RESET INPUT CHECK ANSWER

Object is the always the parent at the top of the tree

parent of Apple is Object

```
public class Apple {
   private String color;

   public Apple() {
      color = "Red";
   }
   public void setColor(String color){
      this.color = color;
   }
   public String getColor() {
      return color;
   }

   public String toString() {
      return "The apple's color is " + color;
   }
```

property

3 **Inheritance 3**
Submitted on Today at 9:13 AM

When a subclass provides a different implementation for a method of its superclass, this is called:

- [ ] method overloading
- [x] method overriding
- [ ] method inheritance
- [ ] method replacement

✓ Correct RESET INPUT CHECK ANSWER

HIDE EXPLANATION

Method overriding allows a subclass to provide its own implementation of a method that's defined in the superclass.

```
public class GrannySmith extends Apple {

   public GrannySmith() {
      super.setColor("Green");
   }
   @Override
   public String toString() {
            return "My granny smith apple is "
                     super.getColor();
   }
```

~~"Green"~~

Refer to the following code for questions 4-6:

```
public class Application{
   psvm(S[] a){
      Apple apple = new Apple();
      apple.setColor = "Green";
```

parent of Apple is Object

```
public class Apple {
   private String color;
   public Apple() {
      color = "Red";
   }
   public void setColor(String color){
      this.color = color;
   }
   public String getColor() {
      return color;
   }
   public String toString() {
      return "The apple's color is " + color;
   }
   public Apple(String color){
      this.color = color;
   }
```

property

Overloading at the class level

Inheritance 3
Submitted on Today at 9:13 AM

When a subclass provides a different implementation for a method of its superclass, this is called:

- method overloading
- method overriding
- method inheritance
- method replacement

Correct RESET INPUT CHECK ANSWER

HIDE EXPLANATION

Method overriding allows a subclass to provide its own implementation of a method that's defined in the superclass.

```
public class GrannySmith extends Apple {

   public GrannySmith() {
      super.setColor("Green");
   }
   @Override
   public String toString() {
            return "My granny smith apple is " +
                  super.getColor();
   }
```

~~"Green"~~

Override a parent's method

Refer to the following code for questions 4-6:

```
public class Application{
   psvm(S[] a){
      Apple apple = new Apple();
      apple.setColor = "Green";
```

Object

toString()

Refer to the following code for questions 4-6:

```
// ##########################
// Alpha.java
// ##########################
public class Alpha {
    public String getOpinion( ) {
        return "Alpha is best!";
    }
}

// ##########################
// Bravo.java
// ##########################
public class Bravo extends Alpha {
    public String getOpinion( ) {
        return "Bravo is best!";
    }
}

// ##########################
// Charlie.java
// ##########################
public class Charlie extends Bravo {

}

// ##########################
// Delta.java
// ##########################
public class Delta extends Alpha {
}
```

Overriding getOpinion

```
public class Demo {
   public static void main (String[] args){
      Alpha alpha = new Alpha();
      System.out.println(alpha.getOpinion());
      Charlie charlie = new Charlie();
      sout (charlie.getOpinion());
      Delta delta = new Delta();
      sout(delta.getOpinion());
      sout(delta.toString());
```

Alpha is best

Bravo is best

Alpha is best

Delta@1b4cd56

Inheritance 4

Inheritance
parent child
super sub
base derived

```java
// Rectangle.java
public class Rectangle {
    private int length;
    private int width;

    // Constructor
    public Rectangle(int length, int width) {
        this.length = length;
        this.width = width;
    }

    public int getArea() {
        return this.length * this.width;
    }
}

// Square.java
public class Square extends Rectangle {

    // Constructor
    public Square(int edgeLength) {
        // What code should be added here?
    }
}
```

1

2 parameters -- must pass in 2 values

What must you add to the Square constructor to make this code work properly?

- `new Rectangle(length, width);`
- `super(edgeLength, edgeLength);` — super stands for the parent
- `Rectangle(edgeLength, edgeLength);`
- `this(length, width);`

square -- all 4 edges are the same length

super() means call the parent constructor

```java
public class Application{
    psvm(S[] a){
        Rectangle rectangle =
            new Rectangle(10, 5);

        sout("The area of the rectangle
        is " + rectangle.getArea());

        Square square = new Square(15);

        sout("The area of the square is " +
            square.getArea());
```

```java
List<Square> myList = new
    ArrayList<>();

myList.add (new Square(10));
myList.add(new Square(4));
myList.add(new Square(6));
myList.add(new Square(10));

Rectangle rectangle2 = new Rectangle(4, 6);
```

Inheritance 7
Submitted on Today at 9:47 AM

Consider the following class definitions: Rectangle.java

```java
// Rectangle.java
public class Rectangle {
    private int length;
    private int width;

    // Constructor
    public Rectangle(int length, int width) {
        this.length = length;
        this.width = width;
    }

    public int getArea() {
        return this.length * this.width;
    }
}
```

Square.java

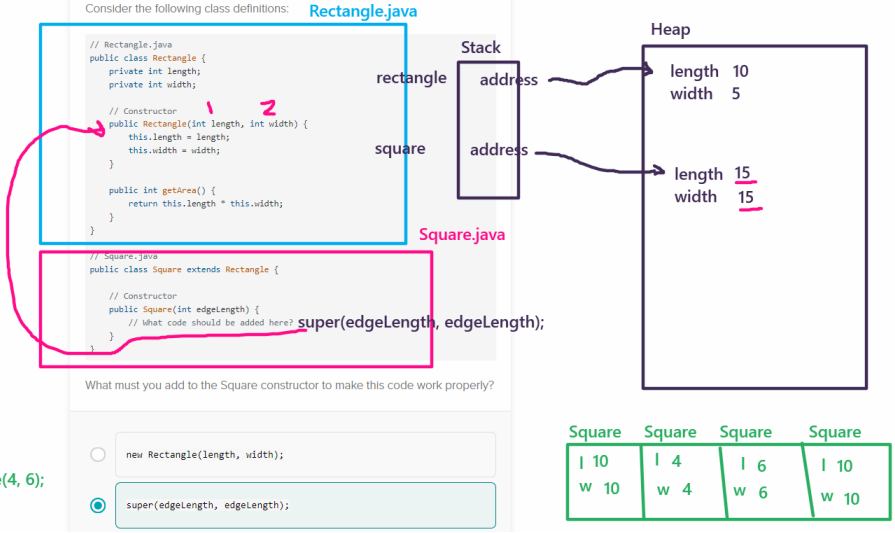
```java
// Square.java
public class Square extends Rectangle {

    // Constructor
    public Square(int edgeLength) {
        // What code should be added here?
    }
}
```

super(edgeLength, edgeLength);

What must you add to the Square constructor to make this code work properly?

- `new Rectangle(length, width);`
- `super(edgeLength, edgeLength);`



Square	Square	Square	Square
l 10	l 4	l 6	l 10
w 10	w 4	w 6	w 10