Chapter 6: Inheritance

EECS 1030

moodle.yorku.ca

State of an object

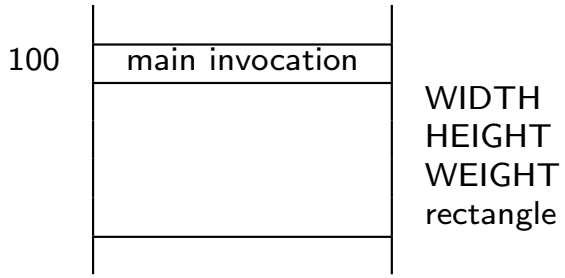
```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80;
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```

Problem

Draw the memory diagram.

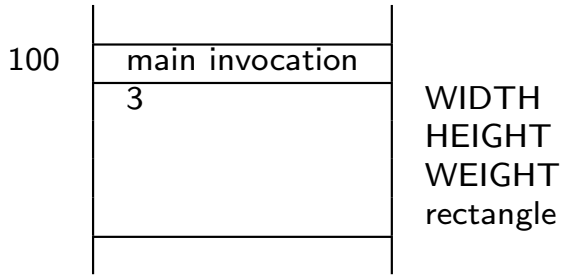
Memory diagram

```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80;
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



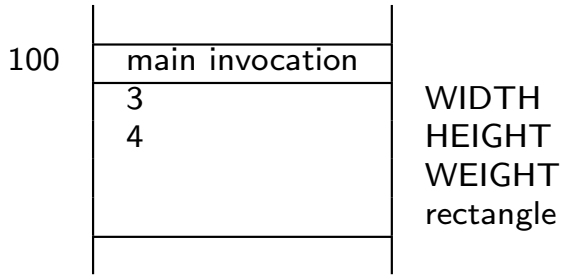
Memory diagram

```
final int WIDTH = 3;
 final   int  HEIGTH = 4;
 final   int  WEIGHT = 80;
GoldenRectangle rectangle  =
   new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



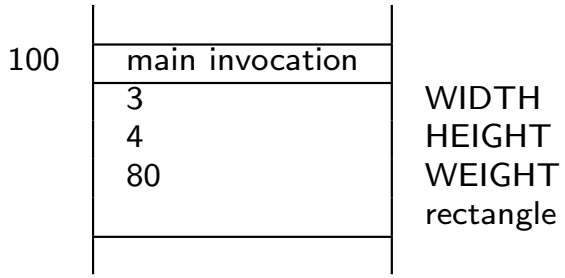
Memory diagram

```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80;
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



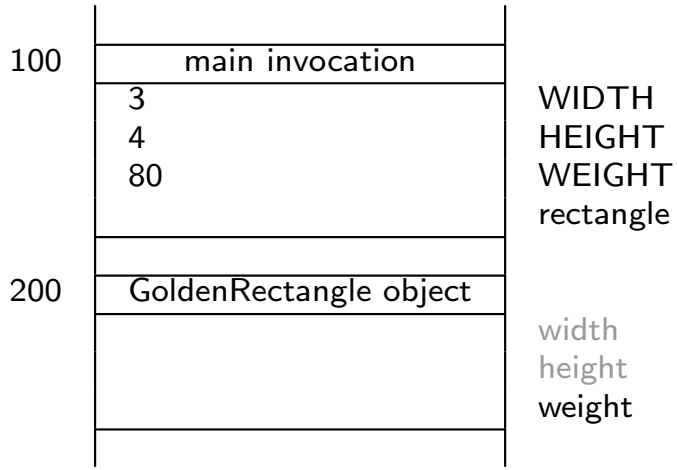
Memory diagram

```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80;
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



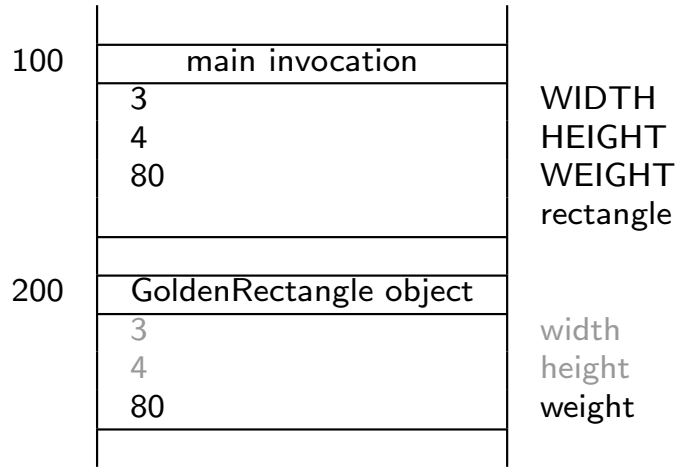
Memory diagram

```
final  int  WIDTH = 3;
final  int  HEIGTH = 4;
final  int  WEIGHT = 80
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



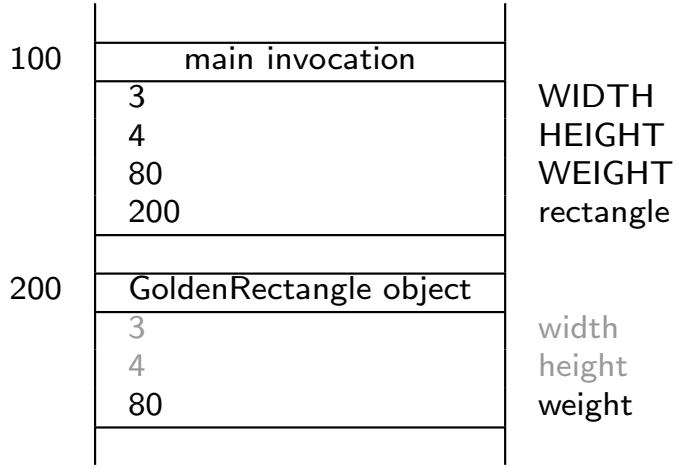
Memory diagram

```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



Memory diagram

```
final   int  WIDTH = 3;
final   int  HEIGTH = 4;
final   int  WEIGHT = 80
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



State of an object

Note

The private attributes width and height of the Rectangle class are part of the state of a GoldenRectangle object, but are *not* inherited.

As a result, the private attributes width and height of the Rectangle class *cannot* be accessed by their name in the GoldenRectangle class.

The extends keyword

To specify that the GoldenRectangle class is a subclass of the Rectangle class, we use the following class header:

```
public class GoldenRectangle extends Rectangle
```

Attributes

Question

Which attributes do we introduce in the GoldenRectangle class? Provide names and types.

Attributes

Question

Which attributes do we introduce in the GoldenRectangle class? Provide names and types.

Answer

```
private int weight;
```

Attributes

Question

Which attributes do we introduce in the GoldenRectangle class? Provide names and types.

Answer

```
private int weight;
```

Note

We do *not* introduce attributes `width` and `height` in the `GoldenRectangle` class, since they are already present in the super class `Rectangle`.

Constructors

Question

Where do we initialize the state of an object?

Constructors

Question

Where do we initialize the state of an object?

Answer

In the constructors.

Constructors

Question

Where do we initialize the state of an object?

Answer

In the constructors.

Question

Which attributes are part of the state of a `GoldenRectangle` object?

Constructors

Question

Where do we initialize the state of an object?

Answer

In the constructors.

Question

Which attributes are part of the state of a `GoldenRectangle` object?

Answer

`width`, `height` and `weight`.

Constructors

Problem

In the constructors of the `GoldenRectangle` class, we have to initialize the attributes `width`, `height` and `weight`. However, we *cannot* access `width` and `height` by their name, that is, *cannot* use `this.width` and `this.height` in the constructors of the `GoldenRectangle` class.

Constructors

Problem

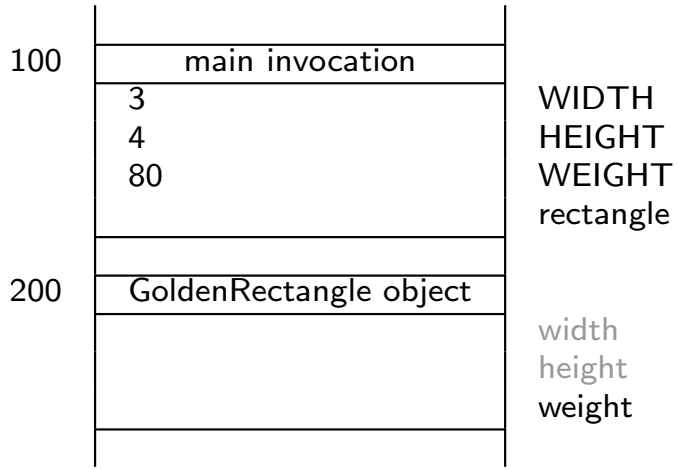
In the constructors of the `GoldenRectangle` class, we have to initialize the attributes `width`, `height` and `weight`. However, we *cannot* access `width` and `height` by their name, that is, *cannot* use `this.width` and `this.height` in the constructors of the `GoldenRectangle` class.

Solution

Delegate to a constructor of the `Rectangle` class.

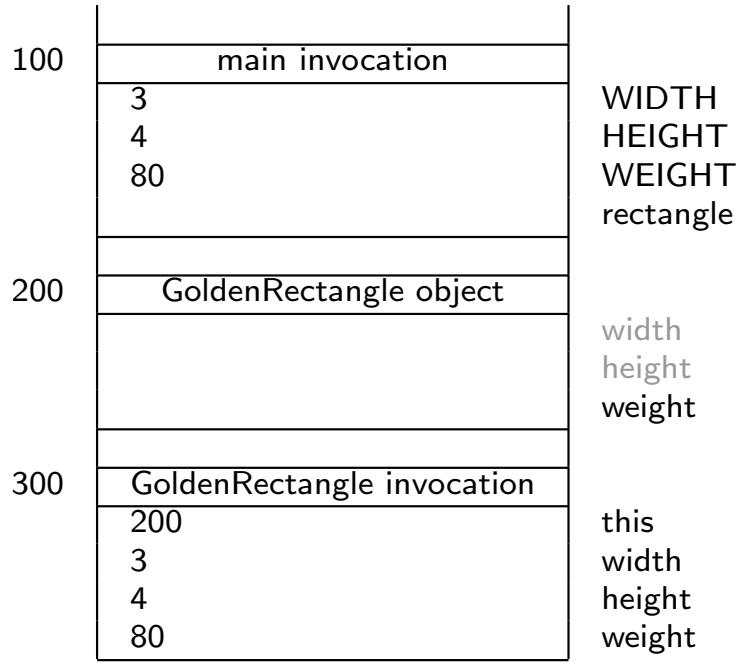
Memory diagram

```
final  int  WIDTH = 3;
final  int  HEIGTH = 4;
final  int  WEIGHT = 80
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



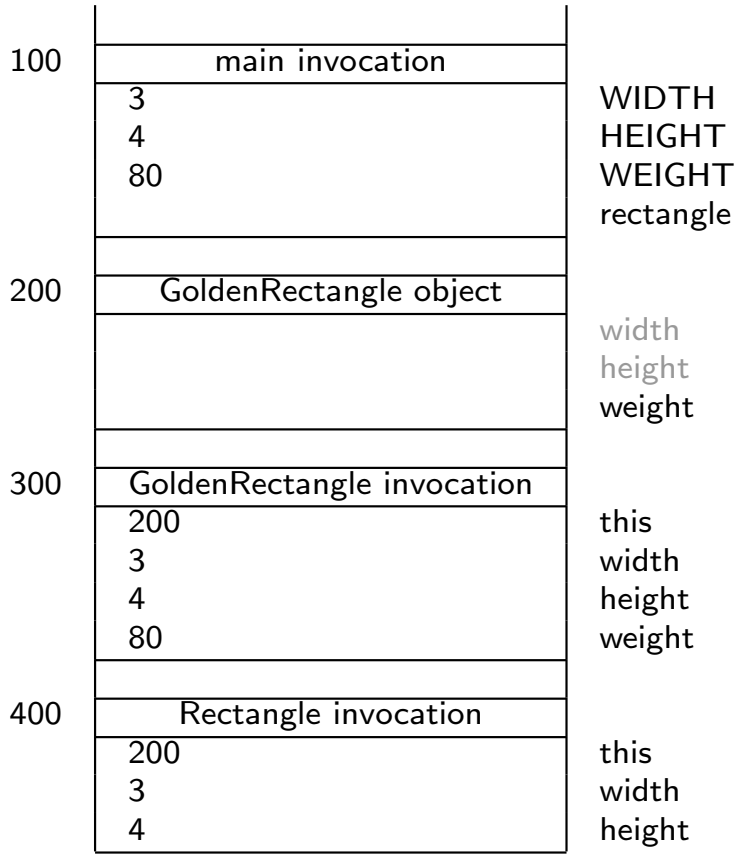
Memory diagram

```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



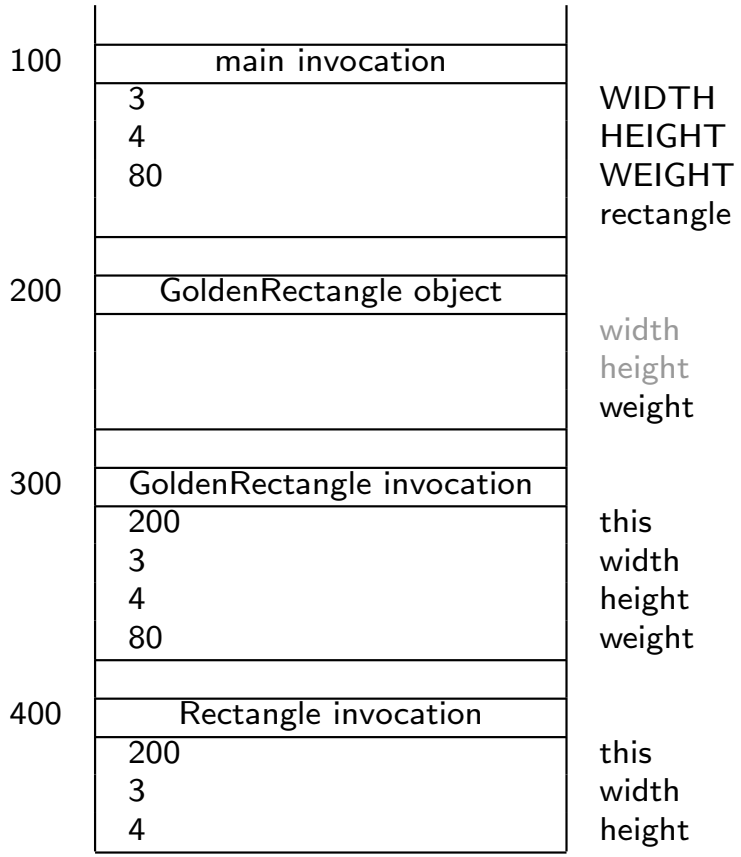
Memory diagram

```
final int WIDTH = 3;
final int HEIGTH = 4;
final int WEIGHT = 80
GoldenRectangle rectangle =
  new GoldenRectangle(WIDTH, HEIGHT, WEIGHT);
```



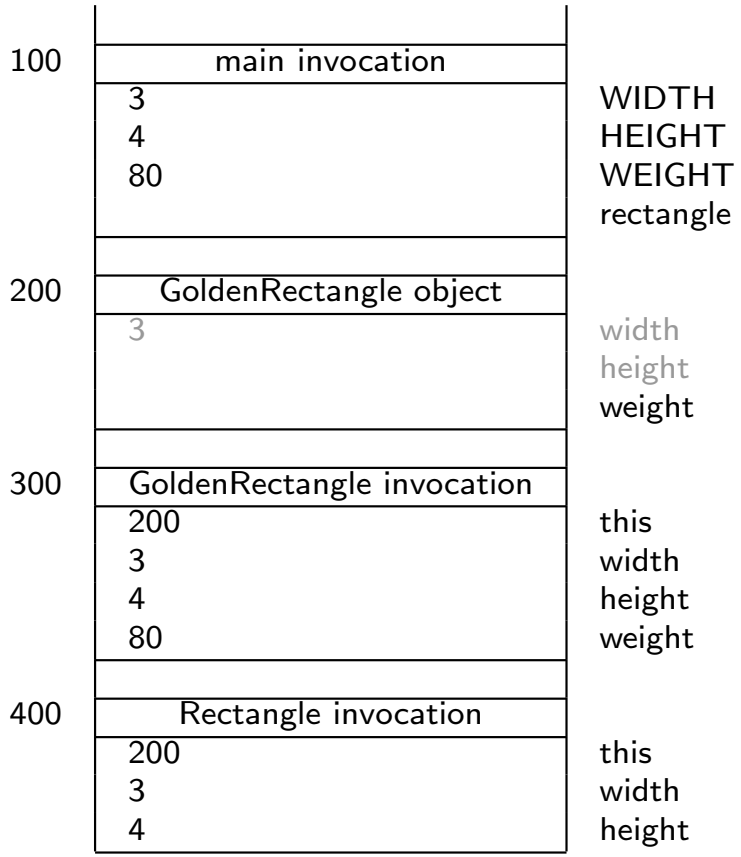
Memory diagram

```
this.width = width;
this.height = height;
```



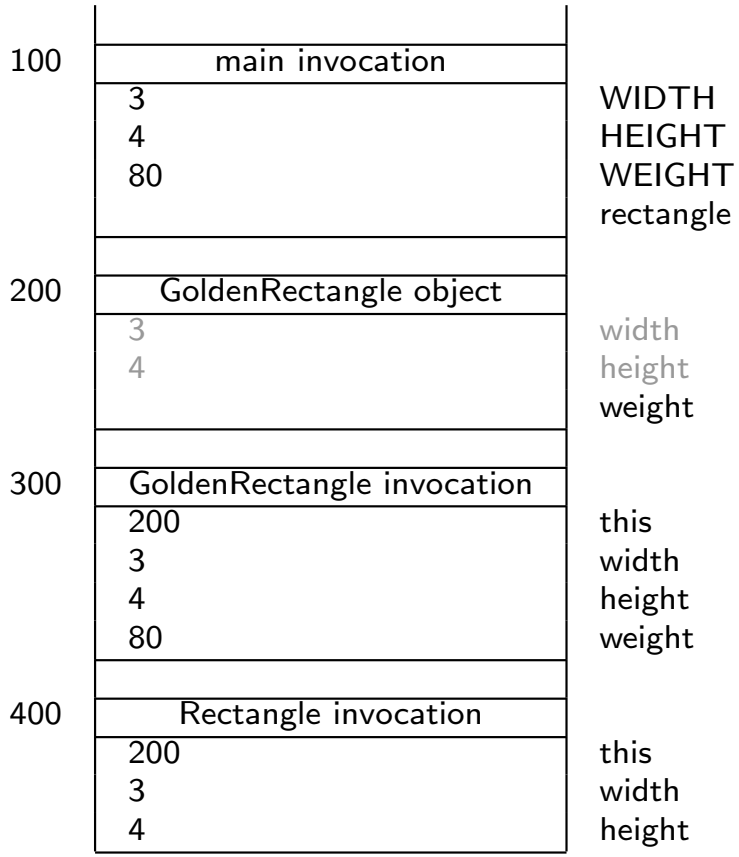
Memory diagram

```
this.width = width;
this.height = height;
```



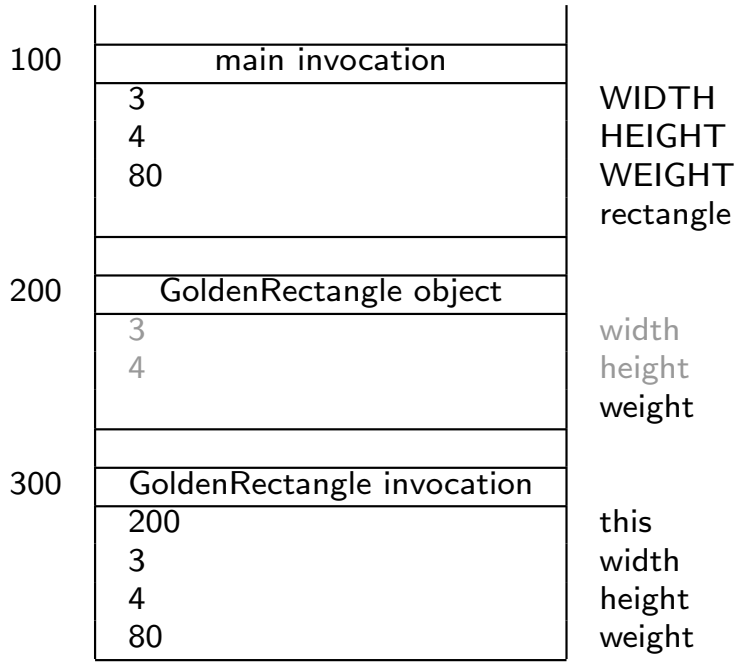
Memory diagram

```
this.width = width;
this.height = height;
```



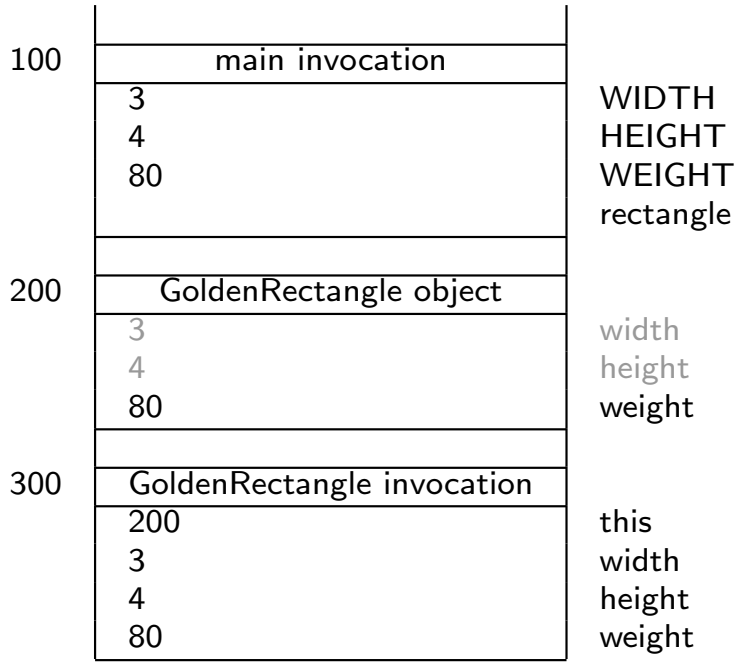
Memory diagram

```
...
this.weight = weight;
```



Memory diagram

```
...
this.weight = weight;
```



Delegation

Question

How do we delegate in the constructor of the `GoldenRectangle` class to the constructor of the `Rectangle` class?

Delegation

Question

How do we delegate in the constructor of the `GoldenRectangle` class to the constructor of the `Rectangle` class?

Answer

Although it may not be the most intuitive syntax, we use

```
super(width, height);
```

Delegation

Question

How do we delegate in the constructor of the `GoldenRectangle` class to the constructor of the `Rectangle` class?

Answer

Although it may not be the most intuitive syntax, we use

```
super(width, height);
```

Note

`super` has an implicit parameter, namely `this`.

Delegation

Question

How do we delegate in the constructor of the `GoldenRectangle` class to the constructor of the `Rectangle` class?

Answer

Although it may not be the most intuitive syntax, we use

```
super(width, height);
```

Note

`super` has an implicit parameter, namely `this`.

Rule

If we delegate to the constructor of the super class, then this has to be done first, that is, `super` needs to be the first statement of the constructor.

Delegation

Problem

Implement the constructor of the `GoldenRectangle` class.

Question

Which methods are inherited by the `GoldenRectangle` class from the `Rectangle` class?

Methods

Question

Which methods are inherited by the `GoldenRectangle` class from the `Rectangle` class?

Answer

compareTo, getArea, getHeight, getWidth, scale, setHeight and setWidth.

Methods

Question

Which methods are inherited by the `GoldenRectangle` class from the `Rectangle` class?

Answer

compareTo, getArea, getHeight, getWidth, scale, setHeight and setWidth.

Question

Which methods are overridden of the `Rectangle` class in the `GoldenRectangle` class?

Methods

Question

Which methods are inherited by the `GoldenRectangle` class from the `Rectangle` class?

Answer

compareTo, getArea, getHeight, getWidth, scale, setHeight and setWidth.

Question

Which methods are overridden of the `Rectangle` class in the `GoldenRectangle` class?

Answer

equals, hashCode and toString.

Question

Which methods of the `GoldenRectangle` class are new?

Methods

Question

Which methods of the `GoldenRectangle` class are new?

Answer

getWeight.

Inherited methods

Nothing needs to be done.

New methods

Problem

Implement getWeight method.

Overridden methods

Problem

Implement equals, hashCode and toString.