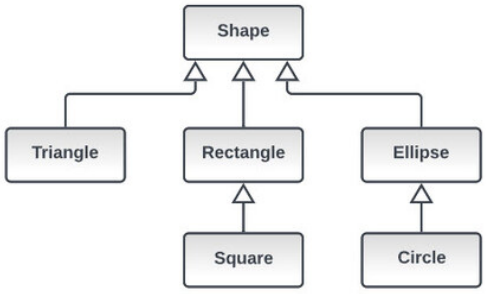
socrative
A showbie PRODUCT

Answer Key
February 13, 2024

Inheritance - Abstract

1. The inheritance relationship can be described by filling in the blank with which phrase?
Square _____ Rectangle



- A "has a"
- ✓ B "is a"
- C "contains a"
- D "looks like a"

Correct: B
Inheritance defines an IS-A relationship. Where the child class inherits all of the properties and behaviors of the parent class. In other words the child class doesn't just have the same functionality as the super class, rather, it can actually be treated as though it were the parent class.

I.E. a Square **IS A** Rectangle and therefore it also **IS A** Shape.

Square square = new Square();

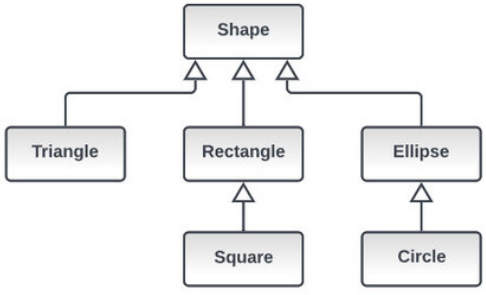
Since a square IS A rectangle the code below is also possible. Even though the new object created in memory is a new square - the variable rectangle ensures that we treat the square as a rectangle.

Rectangle rectangle = new Square();

In this inheritance tree the square a Rectangle inherits from Shape, therefore we can also treat the square just as a shape. Although the Square object in memory has all of the capabilities of a square, the **shape** variable only "knows" what capabilities are defined by shape.

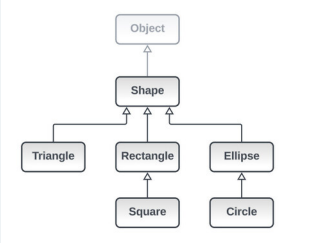
Shape shape = new Square();

2. Rectangle inherits from (extends) which classes?
(Select all that apply)



- A String
- ✓ B Object
- ✓ C Shape
- D Square
- E Triangle
- F Ellipse

Correct: B, C



All classes implicitly extend Object - which means that Object is always the final link in the inheritance chain (i.e. Shape extends Object).

Rectangle → Shape → Object

All other types listed are not in the inheritance chain of the Rectangle class.

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

```
// 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?
    }
}
```

A new Rectangle(length, width);

✓ **B** super(edgeLength, edgeLength);

C Rectangle(edgeLength, edgeLength);

D this(length, width);

E Nothing needs to be added, this code works properly

Correct: B
If the constructor of a parent class accepts input parameters, the child class **is required** provide values for those input parameters.

In this case a Rectangle constructor requires a length and width, so the Square constructor must make a call to the **super** constructor, and pass in 2 integer values.

Since the width and length of a square are the same, the call to super can just pass the edgeLength twice like this: **super(edgeLength, edgeLength)**

4. What syntax is used to have the **Circle** class inherit from **Ellipse**?

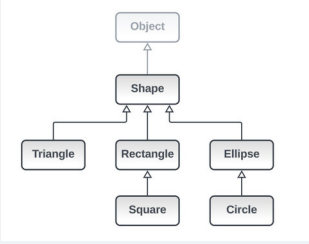
A public class Circle inherits Ellipse

B public class Circle overrides Ellipse

✓ **C** public class Circle extends Ellipse

D public class Circle implements Ellipse

E public class Ellipse extends Circle



Correct: C
In Java, the extends keyword is used to define a class that inherits from another class.

public class Circle **extends** Ellipse
{
// class implementation here
}

5. What is the result of the following class definition?

```
public class Square extends Rectangle, Polygon
{
}
```

✓ **A** Compile Error

B The Square class inherits all properties and methods from both the Rectangle and Polygon classes

C The Square class inherits from the Rectangle class and the Rectangle class inherits from Polygon.

Correct: A
Java does not allow multiple inheritance. It is only possible to extend a single parent class. This code would not compile.

6. The _____ access modifier means that the property or method is available outside of the class.

A visible

B private

C protected

✓ **D** public

E hidden

F inherited

Correct: D
A public variable or method is visible/available both inside and outside the class

7. The _____ access modifier means that the property or method is available only within the class.

A visible

✓ **B** private

C protected

D public

E hidden

F inherited

Correct: B
A private variable or method is visible/available ONLY inside the class

8. The _____ access modifier means that the property or method is available inside the class and to inherited subclasses.

- A visible
- B private
- ✓ C protected
- D public
- E hidden
- F inherited

> **Correct: C**
> A protected variable or method is visible/available inside the class. It is also available to inherited sub-classes. Additionally, in Java it is available to all code that is defined within the same package.

9. A(n) _____ class is a class that cannot be instantiated.

- A super
- ✓ B abstract
- C inherited
- D parent
- E child

> **Correct: B**
> Abstract classes are classes that do not have enough information to be instantiated. You must extend the class and only non-abstract child classes can be instantiated.

10. Methods marked as abstract class **MUST** be overridden

- ✓ T True
- F False

> Methods marked as abstract do not have a method body in the abstract class and must be overridden by the child classes.

11. Which keyword is used in a child class to call the constructor of the parent class.

- A base
- B this
- ✓ C super
- D parent
- E none - the constructor is called automatically

> **Correct: C**
> When the parent/super class has a constructor with parameters, the constructor must be called explicitly to provide values for those parameters.

You call the parent's constructor with the keyword **super**.

```
public class Square extends Rectangle
{
public Square( int side)
{
super(side, side);
}
}
```

note that the call to the super constructor **MUST** be the first call in the constructor of the child.

12. All methods in an abstract class **MUST** be overridden

T True

✓ **F** False

i Only methods marked as abstract must be overridden by the child classes.