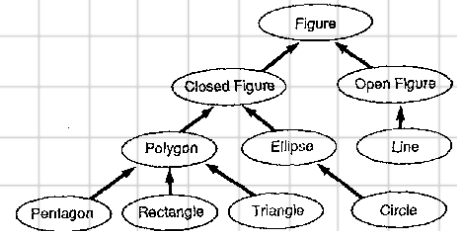


# AP CS: Lesson 20: Inheritance & super. (Superclass)

Name: \_\_\_\_\_ Period: \_\_\_\_\_

## Inheritance: (BJP Chapter 9.1)

- **is-a relationship:** A hierarchical connection where one category can be treated as a specialized version of another.
  - every marketer *is an* employee
  - every legal secretary *is a* secretary



- **inheritance hierarchy:** A set of classes connected by is-a relationships that can share common code.

- **inheritance:** A way to form new classes based on existing classes, taking on their attributes/behavior.
  - a way to group related classes
  - a way to share code between two or more classes
  - One class can *extend* another, absorbing its data/behavior.
- **superclass:** The parent class that is being extended.
  - **subclass:** The child class that extends the superclass and inherits its behavior.
    - Subclass gets a copy of every field and method from superclass

Syntax:

```
public class name extends superclass {
```

Example:

```
public class Secretary extends Employee {  
    ...  
}
```

- **override:** To write a new version of a method in a subclass that replaces the superclass's version.
  - No special syntax required to override a superclass method. Just write a new version of it in the subclass.

## AP CS: Lesson 20: Inheritance & super. (Superclass)

### super. Interacting with the superclass: (BJP Chapter 9.2)

- **Subclasses** can call overridden methods with super  
`super.method(parameters)`

Example:

```
public class LegalSecretary extends Secretary {
    public double getSalary() {
        double baseSalary = super.getSalary();
        return baseSalary + 5000.0;
    }
    ...
}
```

- **Constructors are not inherited.**
  - Subclasses don't inherit the `Employee(int)` constructor.
  - Subclasses receive a default constructor that contains:

```
public Lawyer() {
    super(); // calls Employee() constructor
}
```

- **Calling superclass constructor**

```
super(parameters);
```

Example:

```
public class Lawyer extends Employee {
    public Lawyer(int years) {
        super(years); // calls Employee constructor
    }
    ...
}
```