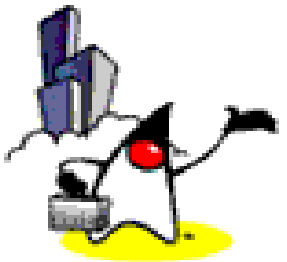
Polymorphism

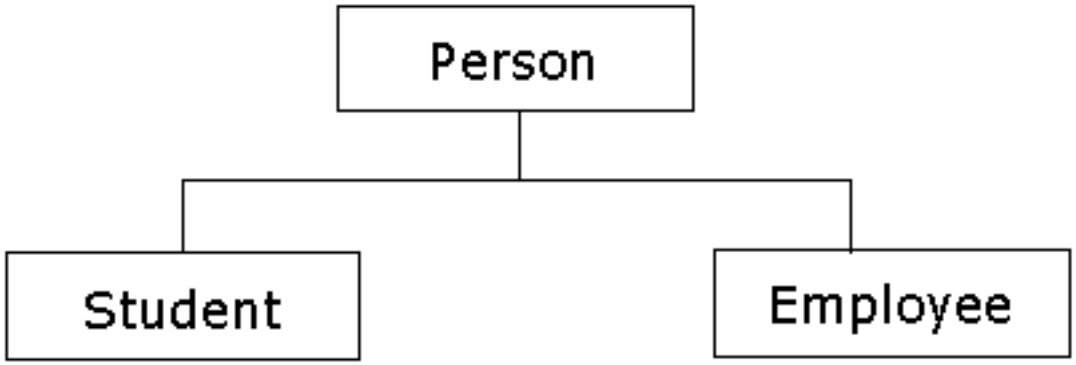


What is Polymorphism?

- Polymorphism
  - The ability of a reference variable to change behavior according to what object instance it is holding.
  - This allows multiple objects of different subclasses to be treated as objects of a single super class, while automatically selecting the proper methods to apply to a particular object based on the subclass it belongs to.

Example: Polymorphism

- Given the parent class **`Person`** and the subclass **`Student`** of the previous examples, we add another subclass of **`Person`** which is **`Employee`**.
- Below is the class hierarchy for that,



Example: Polymorphism

- In Java, we can create a reference that is of type super class to an object of its subclass. For example,

```
public static main( String[] args ) {

    Student studentObject = new Student();
    Employee employeeObject = new Employee();

    Person ref = studentObject; //Person reference points
                                // to a Student object
    String name = ref.getName();
}
```

Example: Polymorphism

- Now suppose we have a getName method in our super class Person, and we override this method in both Student and Employee subclass's

```
public class Student {
   public String getName(){
      System.out.println("Student Name:" + name);
      return name;
   }
}

public class Employee {
   public String getName(){
      System.out.println("Employee Name:" + name);
      return name;
   }
}
```

Polymorphism

- Going back to our main method, when we try to call the **`getName`** method of the reference Person ref, the **`getName`** method of the **`Student`** object will be called.

- Now, if we assign ref to an **`Employee`** object, the **`getName`** method of **`Employee`** will be called.

Example: Polymorphism

```
1   public static main( String[] args ) {
2
3       Student studentObject = new Student();
4       Employee employeeObject = new Employee();
5
6       Person ref = studentObject; //Person ref. points to a
7                                   // Student object
8
9       // getName() method of Student class is called
10      String temp= ref.getName();
11      System.out.println( temp );
12
13      ref = employeeObject; //Person ref. points to an
14                            // Employee object
15
16      //getName() method of Employee class is called
17      String temp = ref.getName();
18      System.out.println( temp );
19  }
```

Polymorphism

- Another example that illustrates polymorphism is when we try to pass a reference to methods as a parameter

- Suppose we have a static method printInformation that takes in a Person reference as parameter.

```
public static printInformation( Person p ){
    // It will call getName() method of the
    // actual object instance that is passed
    p.getName();
}
```

Polymorphism

- We can actually pass a reference of type Employee and type Student to the printInformation method as long as it is a subclass of the Person class.

```
public static main( String[] args )
{
        Student   studentObject = new Student();
        Employee  employeeObject = new Employee();

        printInformation( studentObject );

        printInformation( employeeObject );
}
```