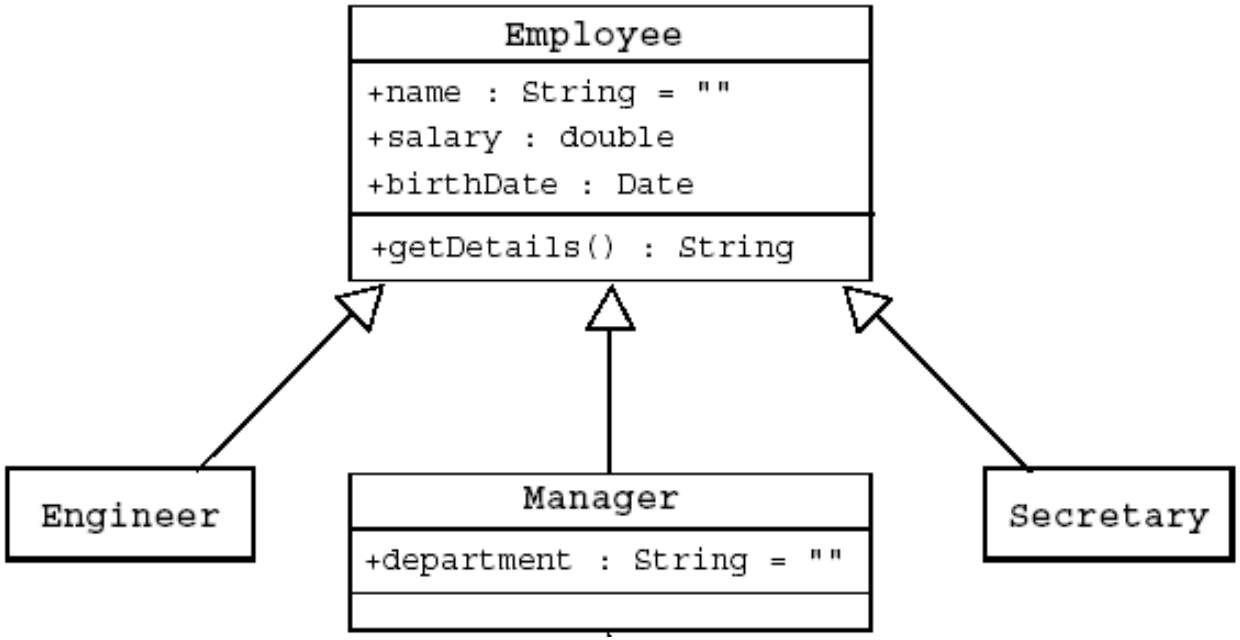
Inheritance

- Inheritance means that a class derives a set of attributes and related behavior from a parent class.
- Benefits of Inheritance:
  - Reduces redundancy in code
  - Code can be easily maintained
  - Extends the functionality of an existing class

Inheritance (Contd.)

- Single Inheritance
  - The subclasses are derived from one super class.
  - An example of single inheritance is as follows:



Inheritance (Contd.)

- Java does not support multiple inheritance.
- Interfaces provide the benefits of multiple inheritance without drawbacks.
- Syntax of a Java class in order to implement inheritance is as follows:

```
<modifier> class <name> [extends
superclass>]
{   <declaration>*   }
```

Access Control

- Variables and methods can be at one of the following four access levels:
  - `public`
  - `protected`
  - `default`
  - `private`
- Classes can be at the public or default levels.
- The default accessibility (if not specified explicitly), is package-friendly or package-private.

Access Control

- Variables and methods can be at one of the following four access levels:
  - `public`
  - `protected`
  - `default`
  - `private`
- Classes can be at the public or default levels.
- The default accessibility (if not specified explicitly), is package-friendly or package-private.

Overriding Methods

- A subclass can modify behavior inherited from a parent class.
- Overridden methods cannot be less accessible.
- A subclass can create a method with different functionality than the parent's method but with the same:
  - Name
  - Return type
  - Argument list

Overriding Methods (Contd.)

- A subclass method may invoke a superclass method using the `super` keyword:
  - The keyword `super` is used in a class to refer to its superclass.
  - The keyword `super` is used to refer to the members of superclass, both data attributes and methods.
  - Behavior invoked does not have to be in the superclass; it can be further up in the hierarchy.

Polymorphism

- Polymorphism is the ability to have many different forms; for example, the Manager class has access to methods from Employee class.
  - An object has only one form.
  - A reference variable can refer to objects of different forms.
  - Java programming language permits you to refer to an object with a variable of one of the parent class types.

    For example:

```
Employee e = new Manager(); // legal
```

Virtual Method Invocation

- Virtual method invocation is performed as follows:

```
Employee e = new Manager();
e.getDetails();
```

  - Compile-time type and runtime type invocations have the following characteristics:
    - The method name must be a member of the declared variable type; in this case Employee has a method called getDetails.
    - The method implementation used is based on the runtime object's type; in this case the Manager class has an implementation of the getDetails method.

Heterogeneous Collections

- Heterogeneous Collections:
  - Collections of objects with the same class type are called homogeneous collections. For example:

```
MyDate[] dates = new MyDate[2];
dates[0] = new MyDate(22, 12, 1964);
dates[1] = new MyDate(22, 7, 1964);
```

  - Collections of objects with different class types are called heterogeneous collections. For example:

```
Employee [] staff = new Employee[1024];
staff[0] = new Manager();
staff[1] = new Employee();
staff[2] = new Engineer();
```

Summary

- In this session, you learned that:
  - Arrays are objects used to group data objects of the same type. Arrays can be of primitive or class type.
  - Arrays can be created by using the keyword `new`.
  - A multidimensional array is an array of arrays.
  - All array indices begin at 0. The number of elements in an array is stored as part of the array object in the `length` attribute.
  - An array once created can not be resized. However the same reference variable can be used to refer to an entirely new array.
  - The Java programming language permits a class to extend one other class i.e, single inheritance.

Summary (Contd.)

- Variables and methods can be at one of the four access levels: `public`, `protected`, `default`, or `private`.
- Classes can be at the public or default level.
- The existing behavior of a base class can be modified by overriding the methods of the base class.
- A subclass method may invoke a superclass method using the `super` keyword.
- Polymorphism is the ability to have many different forms; for example, the Manager class (derived) has the access to methods from Employee class (base).
- Collections of objects with different class types are called heterogeneous collections.