SOFTWARE DESIGN & ANALYSIS (Week-10)

USAMA MUSHARAF

MS-CS (Software Engineering)

LECTURER (Department of Computer Science)

FAST-NUCES PESHAWAR

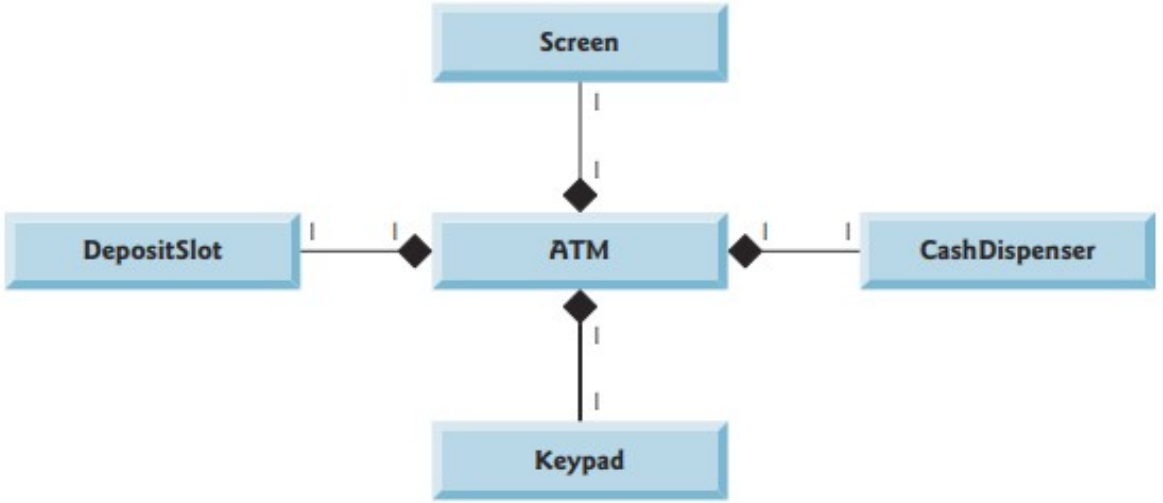
WEEK # 10

- Packages
- Constructors
- Arrays,
- Multi-Dimensional Arrays
- Passing Array to Functions
- Composition
- Inheritance

Composition

A class can have references to objects of other classes as members.

This is called composition.



Lets Code!

Inheritance

- In general inheritance is transfer of characteristics from parent to offspring.
- A child inherit characteristics of its parents, besides inherited characteristics a child may have its own characteristics.
- Inheritance is the process of creating new classes called derived classes from existing classes.
- The class which inherits the properties of other is known as subclass derived class, child class, and the class whose properties are inherited is known as super class, base class, or parent class.
- Keyword “EXTENDS” is used to inherit the properties of base class.

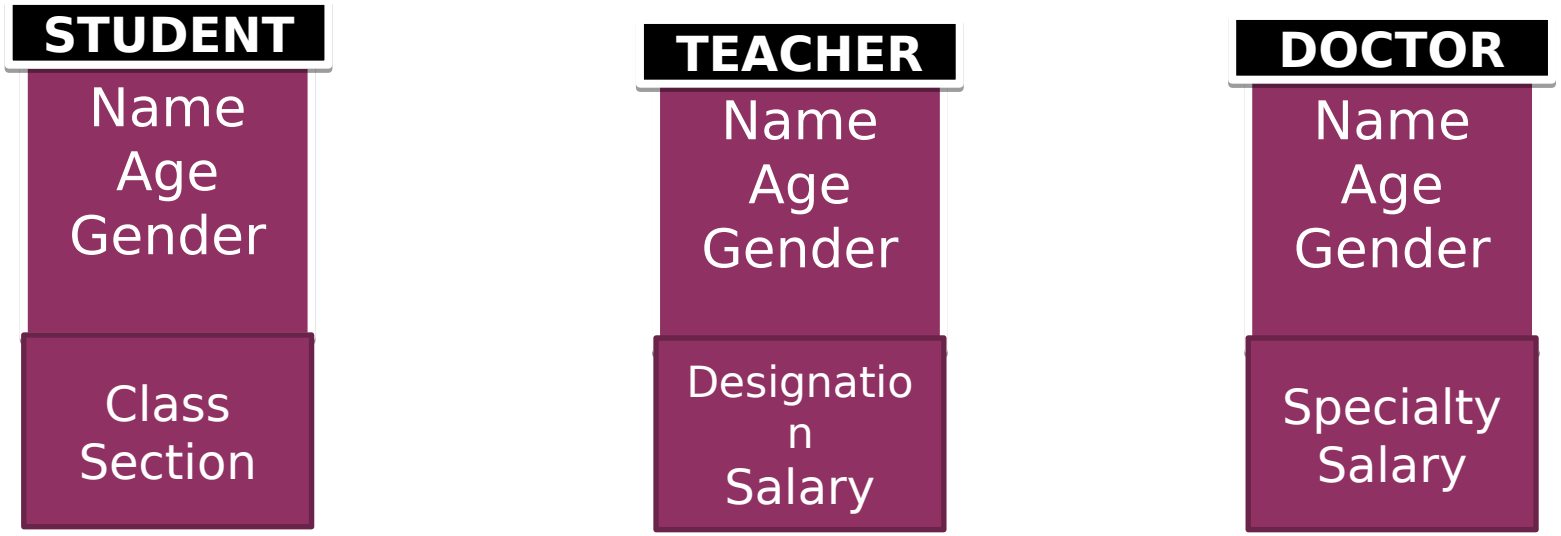
Syntax:

```
class Super
{
data members
member functions
}

class Sub extends Super
{
    data members
    member functions
}
```

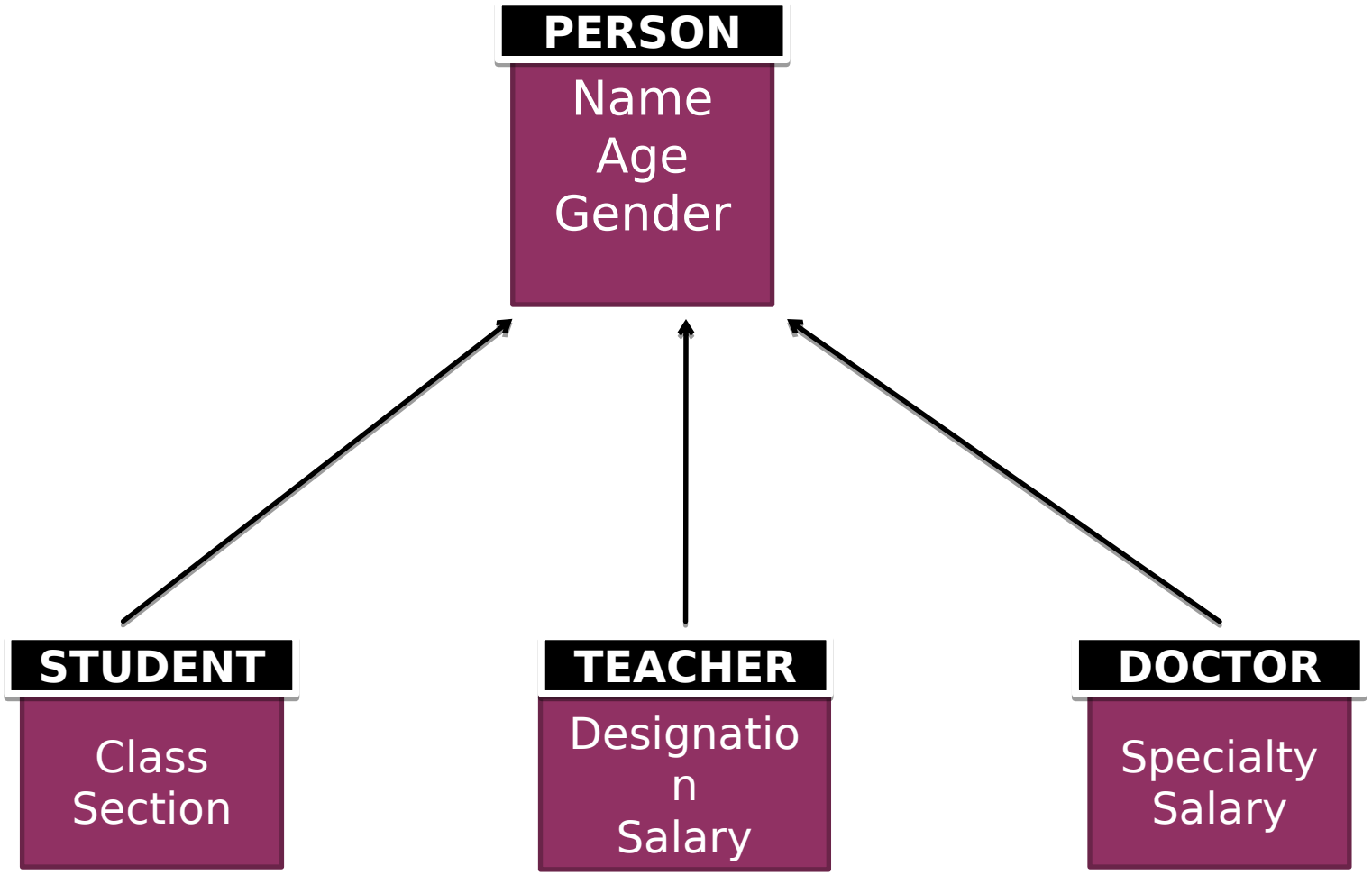
Example:

- Suppose we want to make three classes of teacher student and doctor,
- Each one of then has its own characteristics like.

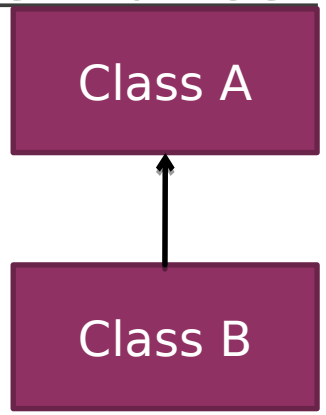


- *Now we have observe that there are some common characteristics in above three classes like (Name, Age, Gender).*

- *So we made a base class named (Person) with common characteristics (name, age, gender), and created some other classes called derived classes (student, teacher , doctor) inherited from the person class.*

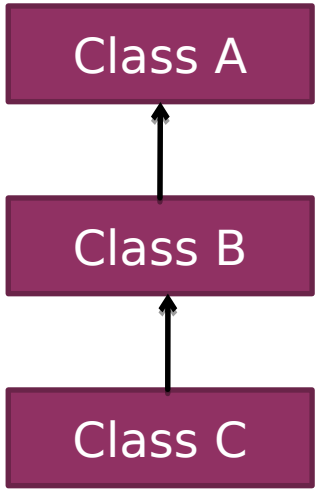

PERSON
Name
Age
Gender
STUDENT
Class
Section
TEACHER
Designatio
n
Salary
DOCTOR
Specialty
Salary

Single Inheritance:



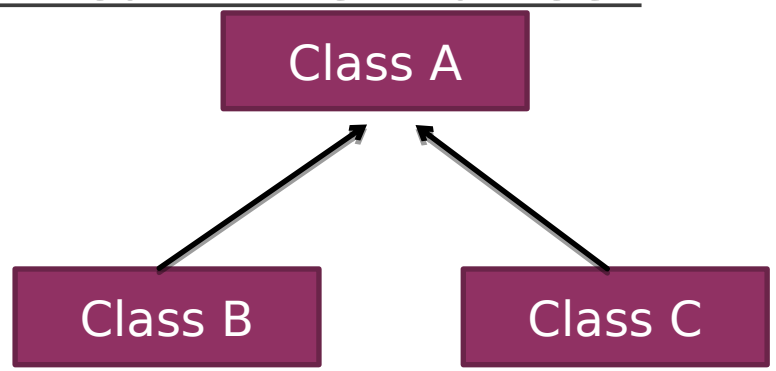

```
Public class A{
……..
}
Public class B extends  A{
……..
}
```

Multi level inheritance:



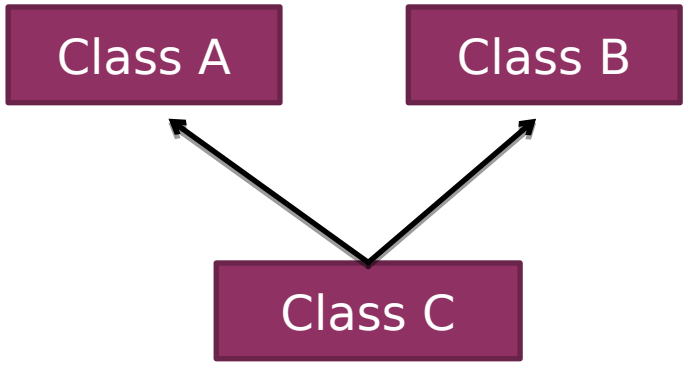

```
Public class A{
……..
}
Public class B extends  A{
……..
}
Public class C extends B{
……..
}
```

Hierarchical Inheritance:



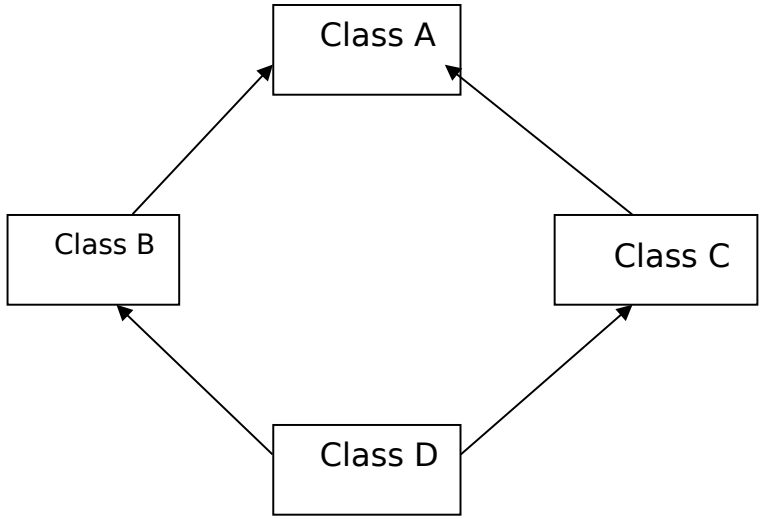

```
Public class A{
……..
}
Public class B extends  A{
……..
}
Public class C extends  A{
…….
}
```

Multiple inheritance:



JAVA DOES NOT SUPPORT MULTIPLE INHERITANCE

Why JAVA does not support Multiple Inheritance ?



Diamond Problem

Classes B and C inheriting from class A.

Class D is inheriting from both B and C implementing multiple inheritance.

Assume both B and C class have a method with the same name & signature then when we invoke that method so compiler will get confused to which method has to process.

This is called diamond problem.

Due to this problem Java designer ignores this ambiguity.

The same feature can be achieved in Java through implementing interfaces.

Super Keyword:

- *It is used to differentiate the members of super class from the members of base class.*
- *It is used to invoke the super class constructor or function from subclass.*

```
public class Dog extends Animal {
public void walk(){
super. move();
System.out.println("Dogs can run and walk");
}
public static void main(String args[]){
Dog obj = new Dog();
obj.walk ();
     }
  }
```

```
public class Animal {

public  void move(){

System.out.println("Animals
Can Move");
     }
  }
```

FUNCTION/METHOD OVERRIDING:

- The existence of a method in a derived class having the same name and signature as the method of base class, then the derived class method overrides the base class method, this is called method overriding.

Function/Method Overriding:

```
public class Dog extends Animal {
public void move(){
System.out.println("Dogs can also Move");
}
public static void main(String args[]){
Animal obj1 = new Animal();
Animal obj2 = new Dog();
obj1.move();
obj2.move();
    }
 }
```

```
public class Animal {

public  void move(){

System.out.println("Animals Can Move");
    }
 }
```

DIFFERENCE BETWEEN METHOD OVERLOADING & OVERRIDING:

- Method overloading is the existence of two or more methods within a same class having same names but different signature.
- While Methods Overriding is the existence of methods in sub class having same name and signature as the base class methods as a result of which the base class methods are overridden.
- Methods Overloading is resolved using static binding in JAVA at compile time,
- While methods overriding is resolved using dynamic binding in java at run time.

HAVE A GOOD DAY!