Polymorphism means “many forms”. In programming terms it means that an object can belong to more than one class. Since each object has a state, behavior (and Identifier), polymorphism allows it to have different abilities. This is done in two ways: upcasting and downcasting. Upcasting means to create an object of a child class(which are the classes used for implementation) but with the reference a parent class(Person):

```
Person p1 = new Student("Laurent", "Bajrami", 24);
```

This allows for runtime polymorphism which means that the methods in the child class are overridden at execution. However, this only allows an object those methods that they have in common and denies access to any exclusive method a child class may have, like the getPayment(), get and setSalary methods. Downcasting –allows for the effect of inheritance to go from child to parent – so that we can use the exclusive methods in the child class.

```
Person p2 = new Staff("Phil", "Anderson", 24, 1000, a1);
```

```
((Staff)p2)
```

Not we can access the getPayment, getSalary and setSalary methods in the Staff Class

```
System.out.println(((Staff)p2).getPayment());
```

This would not have been possible by simply writing: p2.getPayment();