- Code reuse is the most important benefit of inheritance because subclasses inherits the variables and methods of superclass.

19. Important terminology in Inheritance?

- **Class:** the group of objects which have common properties. It is a template or blueprint from which objects are created.
- **SuperClass**: the class being inherited from (or a base class or a parent class).
- **SubClass**: the class that inherits from another class(or a derived class, extended class, or child class).
  - The subclass can add its own fields and methods in addition to the superclass fields and methods.
- **Reusability**: a mechanism which facilitates you to reuse the fields and methods of the existing class when you create a new class. You can use the same fields and methods already defined in the previous class.

20. Difference between Polymorphism and Inheritance

- Like in real world, Inheritance is used to define the relationship between two classes. It is similar to Father-Son relationship. In Java, we have Parent class (also known as super class) and child class (also known as subclass). Similar to the real-world, Child inherits Parents qualities, methods and codes.
  - A child class can reuse all the codes written in Parent class and only write code for behavior which is different than the Parent.
  - Inheritance is actually meant for code reuse.
- On the other hand, Polymorphism is an ability of object to behave in multiple form.
  - It is classified as overloading and overriding.
- By the way, they are actually related to each other, because its inheritance which makes Polymorphism possible, without any relationship between two class. It is not possible to write polymorphic code.
  - Dynamic Polymorphism → Overriding
  - Static Polymorphism → Overloading

21. Difference between method Overloading and method Overriding?

- First and most important difference between overloading and overriding is that,
  - in case of overloading , method name must be the same, but the parameters must be different;
  - in case of overriding , method name and parameters must be same
- Second major difference between method overloading and overriding is that;
  - We can overload method in the same class but method overriding occurs in two classes that have inheritance relationship.
- We cannot override static, final and private method in Java, but we can overload static, final and private method in Java.
- In method overloading , return type can be same or different. In method overriding , return type must be same or covariant type.

22. What is immutable ?

- Immutable means that once the constructor for an object has completed execution that instance can't be altered.
- This is useful as it means you can pass references to the object around, without worrying that someone else is going to change its contents.
  Especially when dealing with concurrency, there are no locking issues with objects that never change.

```
class Foo {
   private final String myvar;
   public Foo (final String initialValue)
   this.myvar = initialValue;
}
   Public String getValue (){
   return this.myvar;
   }
```