extends is for *extending* a class.

implements is for *implementing* an interface

The difference between an interface and a regular class is that in an interface you can not implement any of the declared methods. Only the class that "implements" the interface can implement the methods. The C++ equivalent of an interface would be an abstract class (not EXACTLY the same but pretty much).

Also java doesn't support **multiple inheritance** for classes. This is solved by using multiple interfaces.

public interface ExampleInterface{

public void do();

public String doThis(int number);

}

public class sub implements ExampleInterface{

public void do(){

//specify what must happen

}

public String doThis(int number){

//specfiy what must happen

}

}

now extending a class

public class SuperClass{

public int getNb(){

//specify what must happen

return 1;

}

public int getNb2(){

//specify what must happen

return 2;

}

}

public class SubClass extends SuperClass{

//you can override the implementation

@Override

public int getNb2(){

return 3;

}

}

in this case

Subclass s = new SubClass();

s.getNb(); //returns 1

s.getNb2(); //returns 3

SuperClass sup = new SuperClass();

sup.getNb(); //returns 1

sup.getNb2(); //returns 2

I suggest you do some more research on **dynamic binding, polymorphism and in general inheritance in Object-oriented programming**