Project Proposal

The main purpose of my project is to show a panel with lots of buttons. This buttons can be used to interact with Eddie’s Club. So you can register, unregister, play with applets created by Eddie, take a survey, etc.

**Polymorphism**

Polymorphism can be demonstrated with a minor modification to the Bicycle class. For example, a printDescription method could be added to the class that displays all the data currently stored in an instance.

public void printDescription(){

System.out.println("\nBike is " + "in gear " + this.gear

+ " with a cadence of " + this.cadence +

" and travelling at a speed of " + this.speed + ". ");

}

**Inheritance**

Base on videos by thenewboston in youtube inheritance is the idea of using a superclass to give properties to other classes that describe the super class

For EX:  
public class Bicycle {

// **the Bicycle class has three *fields***

public int cadence;

public int gear;

public int speed;

// **the Bicycle class has one *constructor***

public Bicycle(int startCadence, int startSpeed, int startGear) {

gear = startGear;

cadence = startCadence;

speed = startSpeed;

}

// **the Bicycle class has four *methods***

public void setCadence(int newValue) {

cadence = newValue;

}

public void setGear(int newValue) {

gear = newValue;

}

public void applyBrake(int decrement) {

speed -= decrement;

}

public void speedUp(int increment) {

speed += increment;

}

}

**Abstract**

Mr. Jakob Jenkov has a good website where he explains how Abstracts work.

You declare an abstract class by:

“You declare that a class is abstract by adding the abstractkeyword to the class declaration. Here is an example:

public abstract class MyAbstractClass {

}

That is all there is to it. Now you cannot create instances ofMyAbstractClass. Thus, the following Java code is no longer valid:

MyAbstractClass myClassInstance = new MyAbstractClass(); //not valid

If you try to compile the code above, the Java compiler will generate an error.”

And this is how you use it

public abstract MyAbstractProcess {

public void process() {

stepBefore();

action();

stepAfter();

}

public void stepBefore() {

//implementation directly in abstract superclass

}

public abstract void action(); // implemented by subclasses

public void stepAfter() {

//implementation directly in abstract superclass

}

}

**JVM on iOS, Android OS**

So you can’t put JVM on iOS naturally. You need ADF which is a mobile extension.  [Application Development Framework](http://www.zdnet.com/oracle-announces-mobile-app-development-tools-4010021997/) unveiled on Monday — uses a Java Virtual Machine (JVM) to support the write-once-for-multiple-platforms concept.

Dalvik software is used to run JVM on Android OS

Key Notes

-If you need help go to youtube

- Study

- Download jdk for android and play with it