1. Log in
2. Unit 5
   - Inheritance (overriding and more)
   - Polymorphism
   - Abstract Classes
   - Interfaces

Oct 16-7:52 AM

A reminder...

Method Overloading VS Method Overriding

```
public int addIt(int a){
        return a++;
}
public double addIt(double a) {
        return a+a;
}
public int addIt(int a, int b) {
        return a+b;
}
```

Overloading -

1. The computer makes the decision on which to use when the program is compiled.
2. Compiler compares the method header to see which one works.
3. Compiler checks to see which one can run and find the one to use.
4. This is called Early Binding or Static Binding (done before running).

Nov 13-12:26 PM

A very important difference...

Method Overriding

```
public class myClass1 {
    public int a;
    public myClass1( ) {
        a=5;
    }
    public int addIt(int x) {
        return x+x;
    }
}
```

```
public class myClass2 extends myClass1
    super( );
    public int b;
    myClass2( ) {
        b=3;
    }
    @Override
    public int addIt(int y) {      //use Capital O
        return y*y;
    }
```

Overriding -

1. The computer makes the decision on which to use while it is running.
2. While running, all the methods might work but the choice on which to use is made.
3. The decision is made while the program is running (run-time decision).
4. This is called Late Binding or Dynamic Binding (done "on-the-spot" when needed).

Nov 13-12:26 PM

A new scenario: Banks ... the Account Class

```
public class Account {

    int number;
    String name;
    String password;

    public Account( ){
        number=0;
        name=" ";
        password="XYZ123";
    }
}
```

Nov 13-1:07 PM

A new scenario: Mutator Method enterAccountInfo( )

```
public void enterAccountInfo(int num, String nm, String pw){
    this.number = num;
    this.name=nm;
    this.password=pw;
}
```

Nov 13-1:07 PM

A new scenario: More Account Methods (accessor) ...

```
public int getAccountNumber(){
    return this.number;
}

public String getPassword(){
    return this.password;
}

public String getName(){
    return this.name;
}
```

Nov 13-1:07 PM

A new scenario: Adding a changePassword method ...

```
public void changePassword(){
    Scanner getPW = new Scanner(System.in);

    System.out.println("Enter your new password: ");
    String newPW = getPW.next();

    System.out.println("Re-enter your new password: ");
    String verifyPW = getPW.next();

    if(newPW.equals(verifyPW))
        this.password=newPW;
    else
        System.out.println("Sorry, entries did not match. No PW change!");
}
```

Nov 13-1:07 PM

accountMain class ...

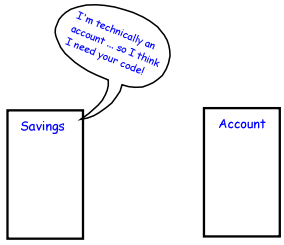
Take some time to create and use the accountMain class to test all aspects of the Account class.

Nov 16-4:14 PM

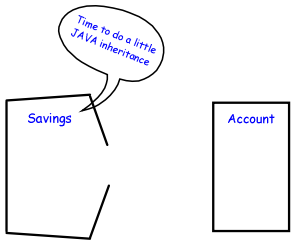
Let's create a Savings subclass ...

1. A savings account is an account.
2. Savings class should inherit the Account class
3. A savings account has "extra" characteristics

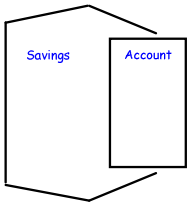
Nov 16-4:14 PM



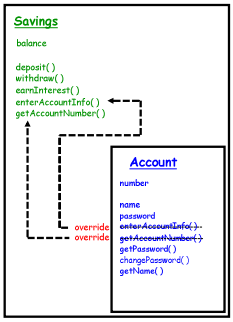
Nov 16-4:17 PM



Nov 16-4:17 PM



Nov 16-4:17 PM



Nov 16-4:21 PM

Coding the Savings class ...

```
public class Savings extends Account{

   double balance;

   public Savings(){
      balance=0;
   }
}
```

Nov 16-4:30 PM

Mutator Methods of the Savings class ...

```
public void deposit(double dep){
   this.balance+=dep;
}

public void withdraw(double wd){
   this.balance-=wd;
}

//no override needed as this is method overloading... diff parameters!

public void enterAccountInfo(int num, String nm, String pw, double bal){
   this.number=num;
   this.name=nm;
   this.password=pw;
   this.balance=bal;
}
```

Nov 16-4:30 PM

Methods of the Savings class ...

```
public void earnInterest(){
   this.balance=(this.balance*1.005);          // 0.5% interest when used
}

@Override                                      //this is override, replaces!
public int getAccountNumber(){
   return this.number+888000000;               //checking starts w/888 & 9 digits
}
```

Nov 16-4:30 PM

accountMain class ...

Take some time to edit and use the accountMain class to test all aspects of the Savings class.

Nov 16-4:18 PM

Things to do ...

1. Wrap Up Unit 5 WS 01-03
2. Work on Unit 5 WS04 More on Inheritance

Oct 16-9:12 AM