OOP Vocabulary

MODULAR - broken into LOGICAL
pieces, called CLASS or OBJECT

No UML diagram for 1 class

Encapsulation =
put things together (in a class)
that belong together, in the right class,
and use PRIVATE, SET..., GET...
for all the variables

INSTANTIATE
... new Student(id , name);
In the STUDENT class, this runs
a CONSTRUCTOR

```
student[27] = new Student(12345,"bob");

student[48] = new Student(2345);

student[123] = new Student("Ronald");
```

We can have multiple constructors
with different PARAMETER LISTS
for different situations

```
new Student(93003,"Smith");
```

ENCAPSULATION requires using
SET to change values, allowing
error prevention.

```
public setName(String name)
{
   boolean found = lookForStudent(name);
   if(found == false)
    {
      studentName = name;
    }
    else
    { output("Name is already registered"); }
}

in STUDENT
public void  returnBook(Loan b)
{
    numBooks = numBooks - 1;
}

static final double PI = 3.1415926;
```

```
Loan thisOne = new Loan(212000,"The Stars");

ST.addLoan(thisOne);

in Student:  collect details

public String showDetails()
{
    String result = "";
    result = result + studentName;

    for(int b=0; b < numBooks; b++)
    {  l = booksBorrowed[b];
       result = result + "/" + l.getBookTitle() + "/" + l.getDate();
    }
    return  result;
}

Two methods with same name: POLYMORPHISM

in Main: find the student

public int showDetails(int ID)
{
      output(ID + ":" + borrowers[ID].getDetails() ) ;
}

INHERITANCE =  extends ANOTHERCLASS

    BOOK
     --  short term books (4 days)
     --  long term books (30 days)
     --  reference cannot be borrowed
     --  pamphlets - free to keep

public shortLoan extends Loan
{
   public double overdueCost()
   {
       Date today = new Date();
       int howLong= today - this.date;
      if ( howLong > 4)
      {  return fine??  }

}

public double longLoan extends Loan
{


}
```