Next Thursday- Midterm 2

Syllabus- a closer look at class and method (whole chapter)

Inheritance (Basic and multilevel hierarchy)

Inner class object:

```
class inn{
  static int arr[];
  int n;
  inn(int arr[], int n){
      this.arr=arr;
      this.n=n;
  }
  static class smallest{
      static double small() {
          double s=arr[0];
          for(int i=0; i<arr.length; i++) {
              if(arr[i]<s) {
                  s=arr[i];
              }
          }
          return s;
      }
  }
  class squareroot{
      double s;
      void sroot() {
          for(int i=1; i<=n; i++) {
              s=Math.sqrt(i);
              System.out.println(s);

          }

      }
  }
```

```
    void analyze() {
        squareroot s= new squareroot();
        s.sroot();
        System.out.println("smallest value is
"+smallest.small());
    }
}

public class innerclassexample {

    public static void main(String[] args) {
        inn i= new inn(new int[] {45,23,12, 1,
5,6,7}, 10);
        inn.smallest si= new inn.smallest();// here
we are creating an object of an inner static class
          si.small();// here I have called the inner
class method by using inner static class object

        inn.squareroot sq= i.new squareroot();// here
we are creating an object of an inner non static
class
        sq.sroot(); // here I have called the inner
class method by using inner non static class object

        //i.analyze();
    }

}
```

final :

```
class f12{
    final int a=45;// here a is a final member and
the value is locked. We wont be able to change the
value of a.
```

```
    final int b=67; // here b is a final member and
the value is locked. We won't be able to change the
value of b
    final int sum() {// here sum is a final method
and we can't modify the element from the sum method.
        return a+b;
    }
}
public class finalexample1 {

    public static void main(String[] args) {
        f12 pf= new f12();
        pf.a=73;X error, since a is a final member of
class f12, we can't change the value of a.
        pf.b=78; X error, since b is a final member
of class f12, we can't change the value of b// TODO
Auto-generated method stub

    }

}
```

Inheritance:

Inheritance is one of the pillars of java besides encapsulation and polymorphism. By using encapsulation, we can encapsulate our members and methods in a class and it is also providing with access control, for example: we use private members so that a different class cant access the member directly. By using polymorphism, we can use same method to execute different actions , for example, method and constructor overloading. In polymorphism, a method /constructor can take many forms, for example , a method can have no parameter, one parameter, two parameters , etc.

Similarly, by using inheritance, a class can inherit another class members and methods. In inheritance, the class which another class uses to inherit members and methods is called superclass. The class which is inheriting is called subclass

Inheritance example:

```
class A12{// Here A12 is the superclass
```

```
    int i,j;
    void showij() {
        System.out.println(i+" "+j);
    }
}
class B12 extends A12{// Here B12 is the subclass of
A12 since B12 is using extends keyword. The format
is: <class+subclass name> extends <superclass name>
    int k;
    void showsum() {
        k=i+j;// Here since B12 is inheriting members
from A12, so the member, i and j of the A12 class can
be used inside B12 class.
        System.out.println(k);
    }
}

public class inheritanceexample {

    public static void main(String[] args) {
        B12 bo= new B12();
        bo.i=12; // we can assign the values to the
members of superclass A12 by using subclass B12
object
        bo.j=35;
        bo.showij();
// We can call the methods from the superclass A12 by
using subclass B12's object
        bo.showsum();
        A12 ao= new A12();
        ao.showsum(); X error, a superclass cannot
use a subclass method.
        ao.k=48; X error, a superclass cannot use a
subclass member.
```

```
        }

    }
```