Sankar Karra
karrasankar@gmail.com

class:

class is a blue print of an object it has **properties(fields), methods, blocks**, etc...

Example:

```
public class AB{
          //block
           Static{
              }
          //property
            int a;
        //method
          void m1(){
           }
   }
```

object:

object is an instance of a class(**copy**).

Example:

```
AB ab=new AB();
```

Data Hiding:

To avoid outside class cannot access our internal data Directly.

by using **private modifier**, we can implement data hiding.

Example:

```
private int a;
```

Abstraction(class) in Java:

1. Hiding Internal implementation
2. must **abstract class** as abstract keyword.
3. **cannot possible to instantiate (Object creation is not possible)**
4. we can allowed to write both abstract and concrete methods(concrete methods nothing but method with body).
5. **it has one zero parameter constructor**.
6. allowed to declare both static and instance variables.

Example:

```
public abstract class{
    //abstract method
   abstract int  m1();
  //concrete method
  int m2(){
 }
}
```

Encapsulation In Java:

Binding of Data and Corresponding methods into a single Unit is called as Encapsulation.

Encapsulation=Data Hiding+Abstraction.

By using private modifier and one pair of public setter and getter methods.

Advantage is security.

Example:

```
public class AB{
    private int a;
         //setter
      public void setA(int a){
          this.a=a;
            }
      //getter
     public int getA(){
        return a;
        }
   }
```

Inheritance(IS-A Relation):

Acquiring Properties from Parent to Child is called as Inheritance.

it is also called as IS-A Relationship

using **extends** keyword we can implement IS-A Relationship

advantage is re usability

java not supported multiple inheritance for to avoid ambiguity problems.

Example: IS-A

```
Public A{
 }
public AB **extends** A{
 }
```

HAS-A RELATIONSHIP:

1. also called as composition or aggregation

2. mostly implemented by using new Operator

3. advantages of HAS-A relationship is Re usability

Example: HAS-A

```
  public A{
    }
 public  AB{
   A a; //HAS-A
  }
```

Polymorphism in java:

Same name with different forms is the concept of Polymorphism

Compile time: Overloading [same method but different parameters]

Runtime   : Overriding

Advantage is flexibility.

Example:

```
  //Overloading
public class AB{
   void m1(){
     }
   void m1(int a){
     }
  }
```

```
//Overriding
public class A{
     void m1(){
        }

}
public class AB extends A{
    @Overriding
     void m1(){
System.out.print("I am came from A class");
       }
}
```

********************************The End***************************