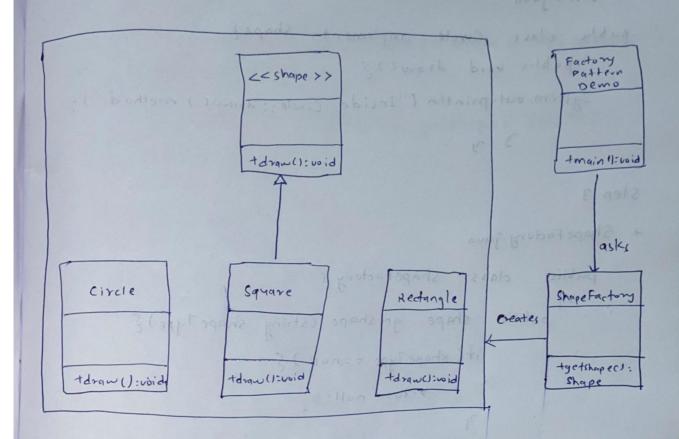
ADDP - Advanced object oriented programming

LAB-1

* Design patterns

1 Factory pattern.



Step 1

Create an Interface clase it (shape Type equals I gnovelose ("Rtc

Shape java () apportant wan nowhere

public interface shapes est square) tiasts void draw ();

Step 2

* Rectangle.java

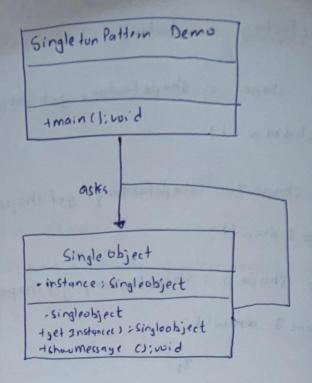
Public class Rectangle implements shape &

public void drawe) {

system.out. println ("Inside Redangle: draw() method").

```
+ Square. Java
     public class square implement
          public void drawe) &
      system.out. println (" Inside square: : draw() method.
              Circle implements Shape {
  * Circleijava
 public class
      public void draw() 5
     system.out. println ("Inside Circle: draw() method.
Step 3
* Shape Factory java
           class Shape Factory &
   public
        public shape getshape (String shape Type) &
               if ( shape Type == null ) {
                    return null;
               ; f (shape Type. equals Ignore (ase ("CIRCLE")) {
                   return new Circle(); 24
               else if ( Shape Type. equals Ignorelase ("RECTANGLE")) &
                   return new Rectangle ();
               else if (shape Type. equals Ignore (ase ("SQUAREI))
                       return new Square();
        return NULL;
                          3 3 ( swarp brow
```

```
* Factory Pattern Demogava
   Public Static void main (String[] args) {
      Shape Factory shape factory = new shape factory ();
     Shape Shape = shape Factory get Shape ("CIRCLE");
     Shapelidraw ();
    Shape Shape 2 = Shape Factory. get Shape ("RELTANGLE");
      Shape 2. draw ();
     Shape Shape 3 = shape Factory get shape ("SOUARE");
       shape 3. draw();
   Step 5
   Output
              Circle: ; draw () method.
      Inside
               Rectangle : : draw() method
     Inside
               Square : draw () method
      Inside
```



Step 1 * single Objectijava

public class SingleObject {

private static SingleObject instance = new SingleObject();

private Single Object() & y

public static SingleObject getInstance() {

peturn instance;

bontom () ways 1: sporation

public void show message ()?

system. out-println ("Hello world");

3 3

step 2

* Single ton Pattern Demoi, and

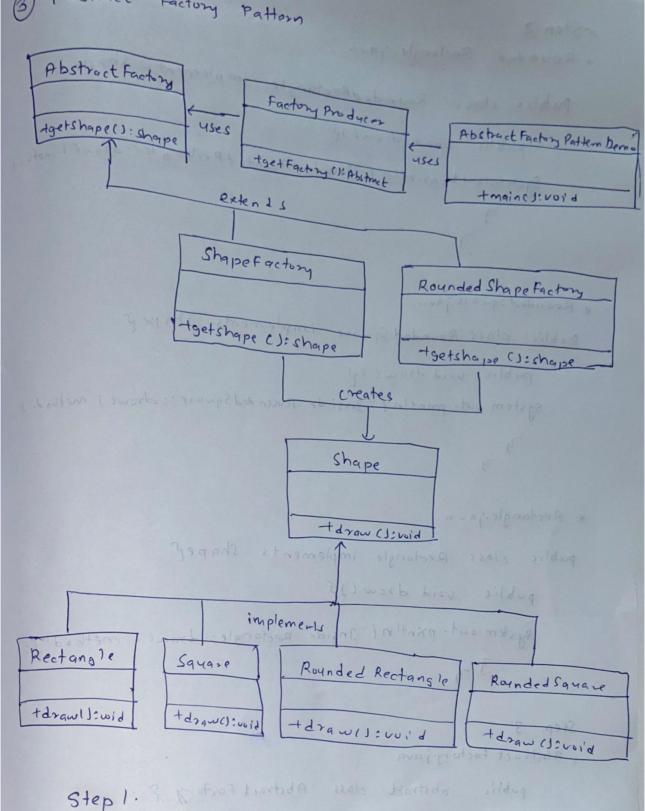
public class Singleton Pattern Denno {

public static void main (String [] args) {

single Object object = Single Object.getInstance();

object.show Message();

3



* Shape java Bande) agade banted

public interface Shape or void draw ();

3 protost met 2 de la desta grutantagariz 12 de vild

```
* Rounded Redangle . java
  Step 2
  public class RoundedRectargle implements Shape p
       public usid drawers
     System.out-println ("Inside Rounded Rectangle: drawe) met
* Rounded Squares qua
 Public class Rounded Square implements Shape of
     public void draw() of
    system. out. printin ("Insi de Roun dod Square: drawe) methodig
* Rectangleijava
public class rectangle implements Shaper
     public void draw () {
     system.out. println ("Inside Rectangle: draw() method");
 Step 3
* Abstract Factory java
      public abstract class Abstract Factory &
          abstract shape getshape (String shape Type);
step 4.
 * Shapefactory. java
    public class shapefactory extends Abstractfactorys
     public shape get shape (String shape Type)
         if (shape Type, equals Ignore (ase ("RECTANGLE"))
```

return non Redan 1

```
Justelase ("SOUARE"))
          setum new Square (1.
          seturn null;
Pourt * Rounded Shape Factory java
   public class Rounded Shape Factory extends Abstract Factory
       public shape getshape (string shapeType)
        if (shape Type, equals Ignore (ase ("RECTANGLE"))
               return new Rounded Rectangle():
          else if CshapeType, equals Ignore(ase ("SQLARE")
              retyrn new Rounded Square();
               seturn NULL;
   Step 5
 * Factory Prolycerigana
      public class Factory Producer &
public Static Abstract Factory get Factory (buolean nounded)
        & if (munded)
           Leturn new Rounded Shape Factory();
                 E seturn new Shapetactory (); y zy zy
```

```
Step 6
```

* Abstract Factory Pattern Demo java

public class AbstractFactory Pattern Demos public static void main (String [] angs)

5

Abstractfactory shape Factory = factory Produces. gatachylle

Shape I shape to shape factory get shape ("RECTANGLED; shape I draw ();

Shape Shape 2 = shapefactory. get shape (" SQUARE").

([13] WAY 3 3 day 2 7 3 would to Cube (Elmin (E.))

4

Step 7 output

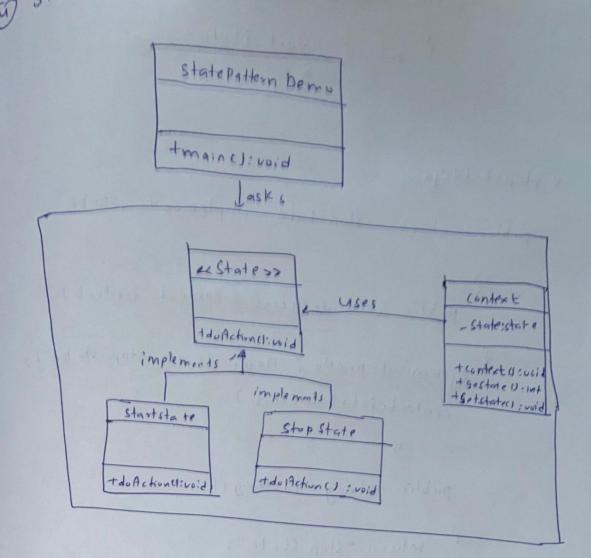
Inside Rectangle :: draw() method

Inside Square : draw () method

: 110 w aruter

3 4 345

Feethern Produces java



Step 1.

* State java

public interface State

E public void do Action (Context context),

Step 2

* Start State gava

public class Startstate implements State

Epublic void du Action (Context context)

Systemout. println ("Player is in start state").

context. set State (+his);

```
public string to String ()
            f return "start state".
* Stopstate : java
 public class Stopstate implements State
         public void do Action (context context)
       System.out. println ("Player is in stop stabe");
         context. set state (this)
         public string tostring ()
        E return "Stop State";
* (ontextigana
    public class context
   E private State state;
        public (ontext ()
        € state = null;
  public void set State (state state)
        this state ;
       public state get state()
         { return state;
```

```
* State Pattern Demojava
public class state Pattern Demo
    public static void main (string[] angs)
     Context context = new context();
   StartState startstate = new Startstate();
      Start State, do Action (context).
 System.out. println (context. getstate 1). to string ());
  Stop State stopState = new stopState();
      Stopstate, do Action (context);
System, out, println (contextiget State (). tostring ());
             3 7
 Step 5.
   output
    player is in start state
    Start State
     player is in stop state
      Stop state
```