Abstract/Final Keywords CODE:

```
package honorsBootcamp;
abstract class demo
{
    abstract void fun();
    abstract void temp();
    void printing(){
        System.out.println("Print Function
Called");
    };
}
abstract class test extends demo{
    void fun()
    {
        System.out.println("Fun method
overriden");
    }
}
class corona extends test{
    void temp()
    {
        System.out.println("temp method
overriden");
    }
}
class testingfinal{
    final void fun(){
        System.out.println("Fun Method");
    }
    final void fun(int n)
    {
        System.out.println(n+ " Parametrized Fun
Method");
    }
```

```
}
//class inheritingfinal extends testingfinal{
//  void fun(){
//      System.out.println("Overridden fun");
//  }
//}
final class yepp{
    void printit(){
        System.out.println("I'm Final");
    }
//  abstract void printto(); cannot have
abstract methods in final class
}

//class yoyo extends yepp{ //Cannot inherit
final class
//
//}

class gocorona{//to test final variables
    final int i=5;//assigning value while
declaring
    final int j;
    //gocorona(){}; cannot use default
constructor as no value will be assigned to j
    gocorona(int j)
    {
        this.j=j;//can assign value in a
constructor
    }
    void trytochange()
    {
        //i=10; cannot assign value to final
variable
```

```
        //j=5; cannot assign value in a normal
method
    }
}
class chhotacorona extends gocorona
{
        chhotacorona(){
            super(5);
        };
        chhotacorona(int i)
        {
            super(i);
        }
        void trychanging(){
            //i=10; cannot change value of final
variable even in dubclass
        }
}
public class abstractstudy {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        corona obj=new corona();
        obj.temp();
        obj.fun();
        obj.printing();
        testingfinal tf=new testingfinal();
        tf.fun();
        tf.fun(123);
        yepp y=new yepp();
        y.printit();
        gocorona gc=new gocorona(20);
        System.out.println("Value of i: "+ gc.i);
        System.out.println("Value of j: "+ gc.j);
    }
```

```
}
```