FACTORY DESIGN PATTERN

PROFESSION INTERFACE:

```
package org.example.Demo;

public interface Profession {
    void print();
}
```

DOCTOR CLASS:

```
package org.example.Demo;

public class Doctor implements Profession{
    @Override
    public void print() {
        System.out.println("i am a doctor");
    }
}
```

ENGINEER CLASS:

```
package org.example.Demo;

public class Engineer implements Profession{
    @Override
    public void print() {
        System.out.println("i am a engineer");
    }
}
```

TEACHER CLASS:

```
package org.example.Demo;

public class Teacher implements  Profession{
    @Override
    public void print() {
        System.out.println("i am a teacher");
    }
}
```

FACTORY CLASS :

```
package org.example.Demo;

public class ProfessionFactory {

    public Profession getProfession(String typeOfProfession){
        if(typeOfProfession==null){
            return null;
        }
```

```
        if(typeOfProfession.equalsIgnoreCase("Doctor")){
            return new Doctor();
        } else if (typeOfProfession.equalsIgnoreCase("Engineer")) {
            return new Engineer();
        } else if (typeOfProfession.equalsIgnoreCase("Teacher")) {
            return new Teacher();
        }
        return null;
    }

}
```

MAIN CLASS:

```
package org.example.Demo;



public class A {
public static void main(String[] args) {
ProfessionFactory professionFactory = new ProfessionFactory();
    Profession doctor = professionFactory.getProfession("Doctor");
    doctor.print();
}
}
```