**Single Responsibility Principle(SRP)**

**there should never be more than one reason for a class to change. A class should be focused on a single functionality**

example 1:

**public class Resume {**

**private String candidateName;**

**private long mobileNumber;**

**private String emailId;**

**private String technology;**

**private Integer yearsOfExperience;**

**public Resume() {}**

**public String getTechnology() {**

**return technology;**

**}**

**public void setTechnology(String technology) {**

**this.technology = technology;**

**}**

**public Integer getYearsOfExperience() {**

**return yearsOfExperience;**

**}**

**public void setYearsOfExperience(Integer yearsOfExperience) {**

**this.yearsOfExperience = yearsOfExperience;**

**}**

**Getter Setters for name,mobile,emailID**

**}**

**public class ResumeRepository {**

**Resume resumes[];**

**public RepositoryView(resumes) {**

**this.resumes = resumes;**

**}**

**public void filterResumeByTechnology(String tech) {**

**for(Resume r: Resumes)**

**{**

**if(resume.getTechnology().equals(tech)**

**{**

**System.out.println(resume.getCandidateName() +resume.getMobile() + resume.getEmail());**

**}**

**}**

**}**

**public void filterResumeByExperience(int years) {**

**for(Resume r: this.Resumes)**

**{**

**if(r. getYearsOfExperience()>=years)**

**{**

**System.out.println(r.getCandidateName() +r.getMobile() + r.getEmail());**

**}**

**}**

**}**

**}**

Class DriverApp

{

Psvm()

{

Resume r1=new Resume();

Resume r2=new Resume();

Resume r1=new Resume();

Resumes resumes[]={r1,r2,r3};

ResumeRepository rr=new ResumeRepository(resumes);

rr. **filterResumeByTechnology(“springboot”);**

rr. **filterResumeByExperience(5);**

}

}

Example 2:

public class TextManipulator {

private String text;

public TextManipulator(String text) {

this.text = text;

}

public String getText() {

return text;

}

public void appendText(String newText) {

text = text.concat(newText);

}

public String findWordAndReplace(String word, String replacementWord) {

if (text.contains(word)) {

text = text.replace(word, replacementWord);

}

return text;

}

public String findWordAndDelete(String word) {

if (text.contains(word)) {

text = text.replace(word, "");

}

return text;

}

public void printText() {

System.out.println(textManipulator.getText());

}

}

public class TextPrinter {

TextManipulator textManipulator;

public TextPrinter(TextManipulator textManipulator) {

this.textManipulator = textManipulator;

}

public void printText() {

System.out.println(textManipulator.getText());

}

public void printOutEachWordOfText() {

System.out.println(Arrays.toString(textManipulator.getText().split(" ")));

}

public void printRangeOfCharacters(int startingIndex, int endIndex) {

System.out.println(textManipulator.getText().substring(startingIndex, endIndex));

}

}

Class DriverApp

{

Psvm()

{

TextManipulator string1=new TextManipulator(“Mahendra Singh ”);

String1. appendText(“Dhoni”);

String1. findWordAndReplace(“Singh”,”Kumar”);

TextPrinter t1=new TextPrinter(string1);

T1. printText();

}

}