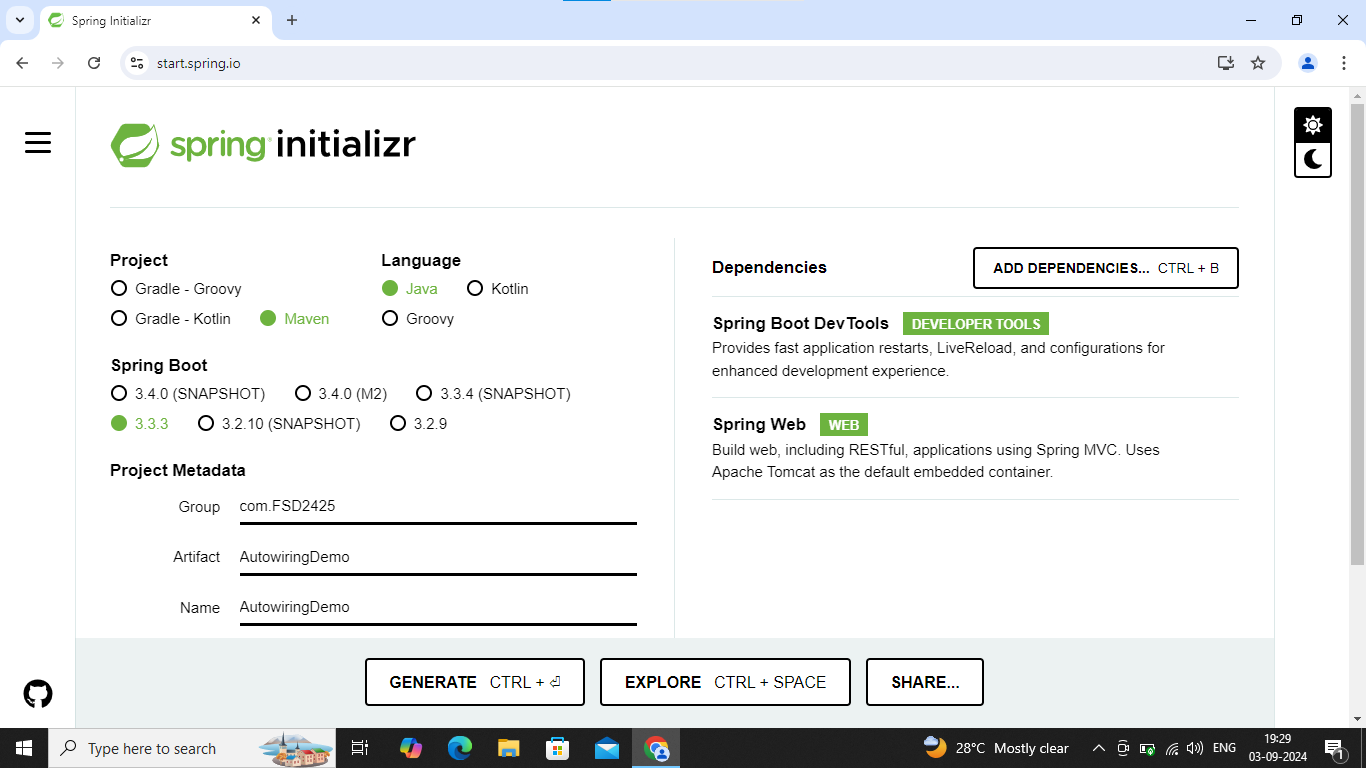
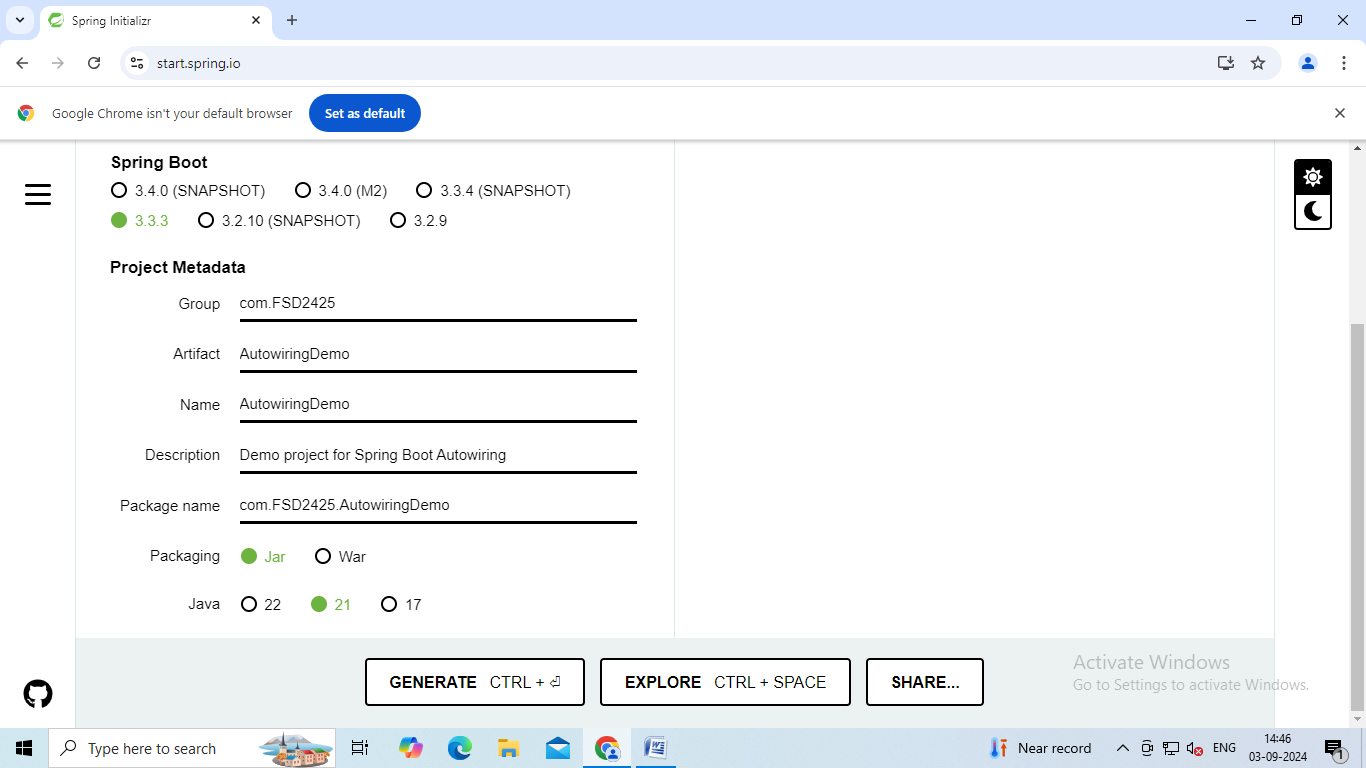
**2. Demonstrate Autowiring by name, by type, and bean scope in spring boot.**

Step1: To set project configuration🡪Go to browser🡪Search **start.spring.io**

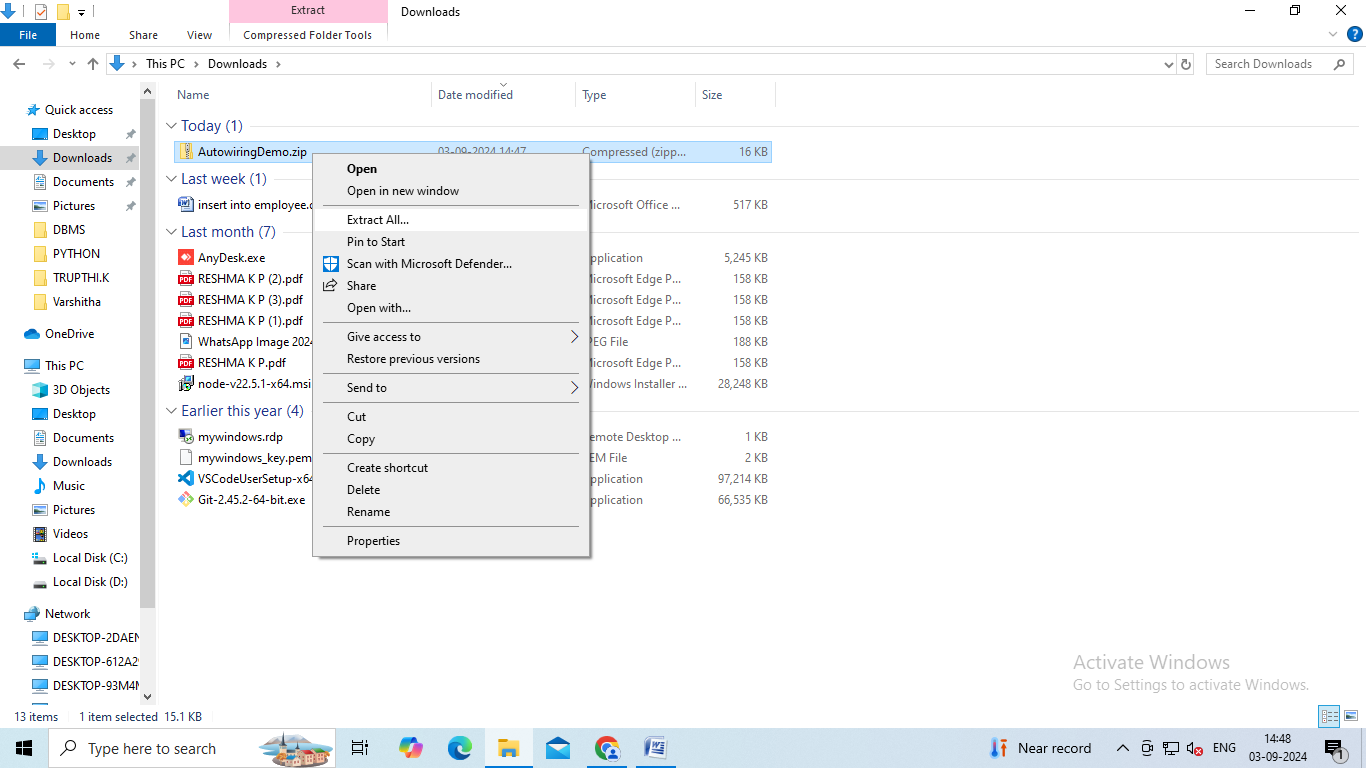
Step2: In spring initialize🡪Select Project-Maven🡪Language-Java🡪Spring Boot-3.3.3🡪 Group-com.FSD2425🡪Artifact-AutowiringDemo🡪Description-Demo project for spring boot Autowiring🡪Packaging-Jar🡪Java-21.

Dependencies🡪ADD DEPENDENCIES🡪Select-Spring Boot Dev Tools & Spring Web🡪 Click on Generate.



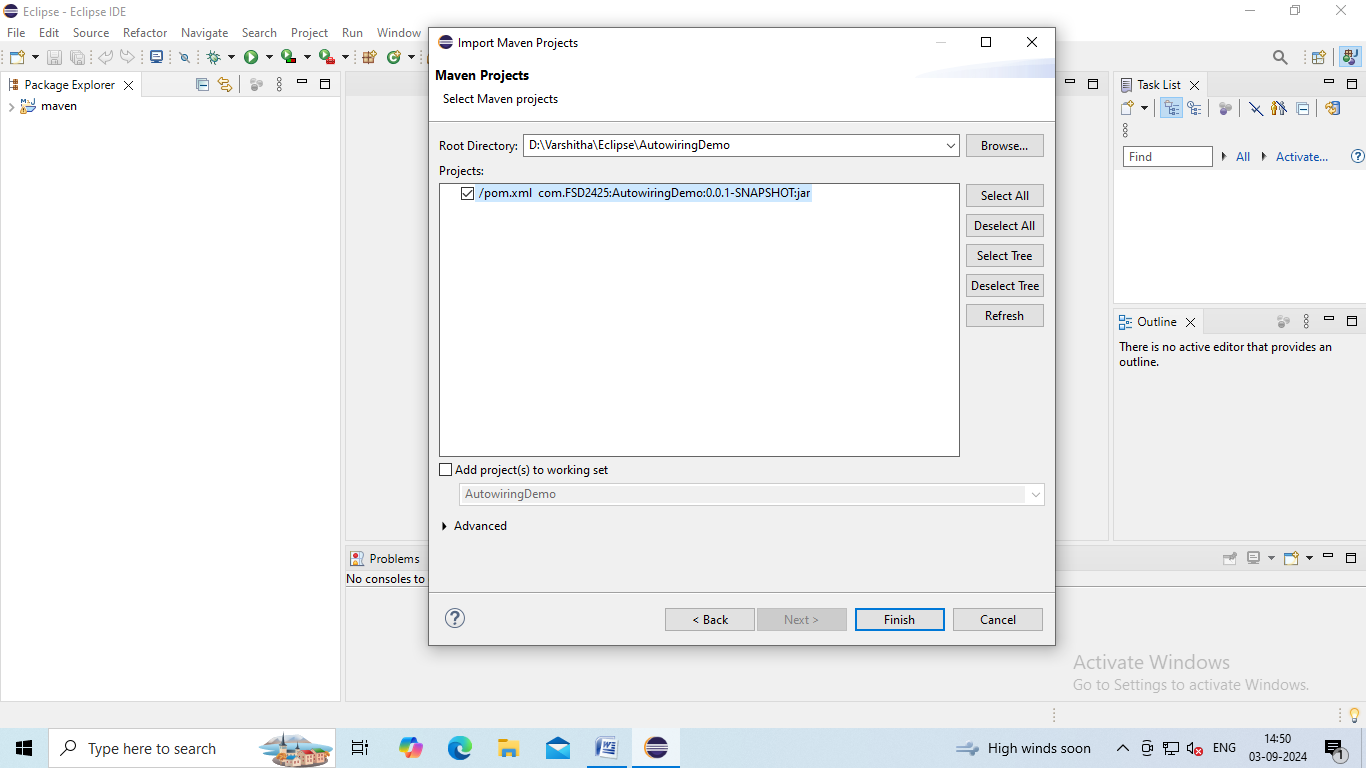


Step2: Next extract your zip file to workspace.



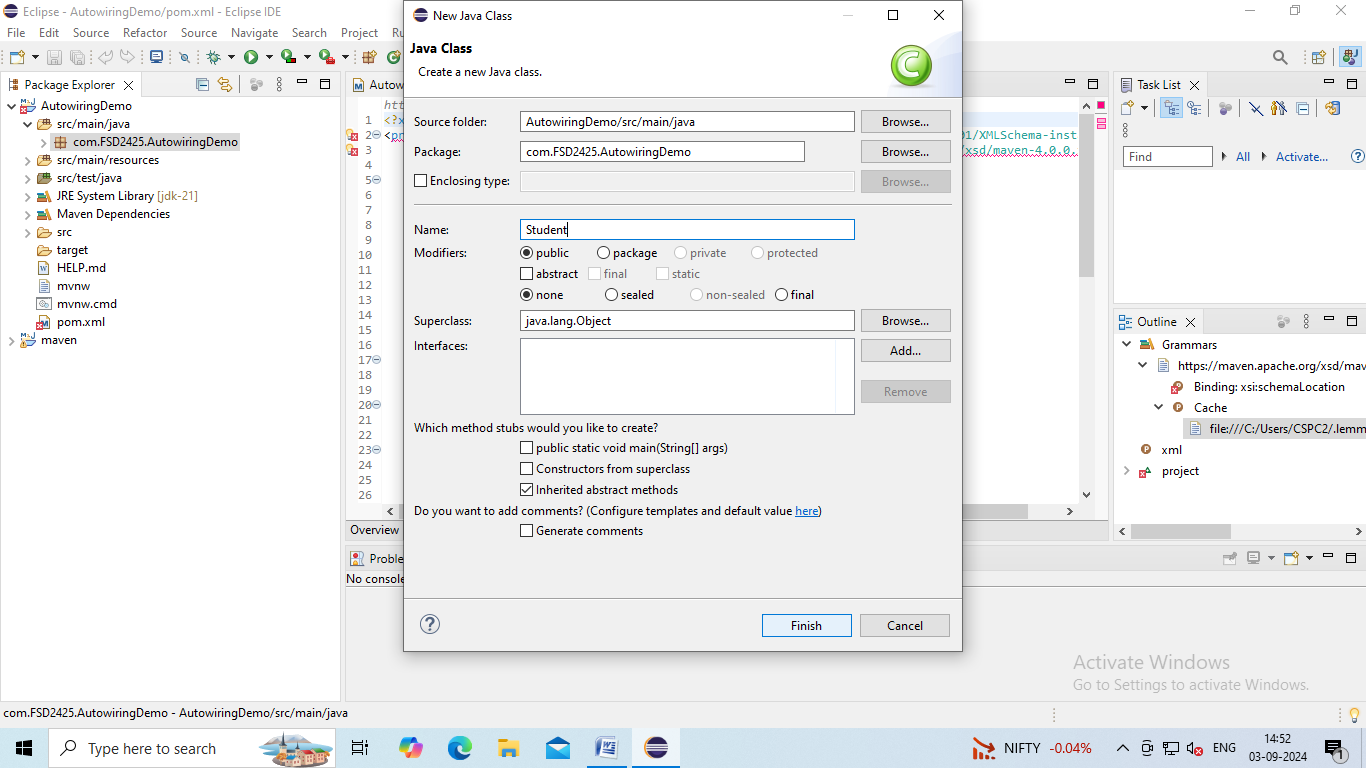
Step3: Go to Eclipse🡪File🡪Import🡪Maven🡪Existing maven🡪Click on Browse and select the folder from where you extracted🡪Tick mark on pom.xml🡪Click on Finish.

Save the pom.xml



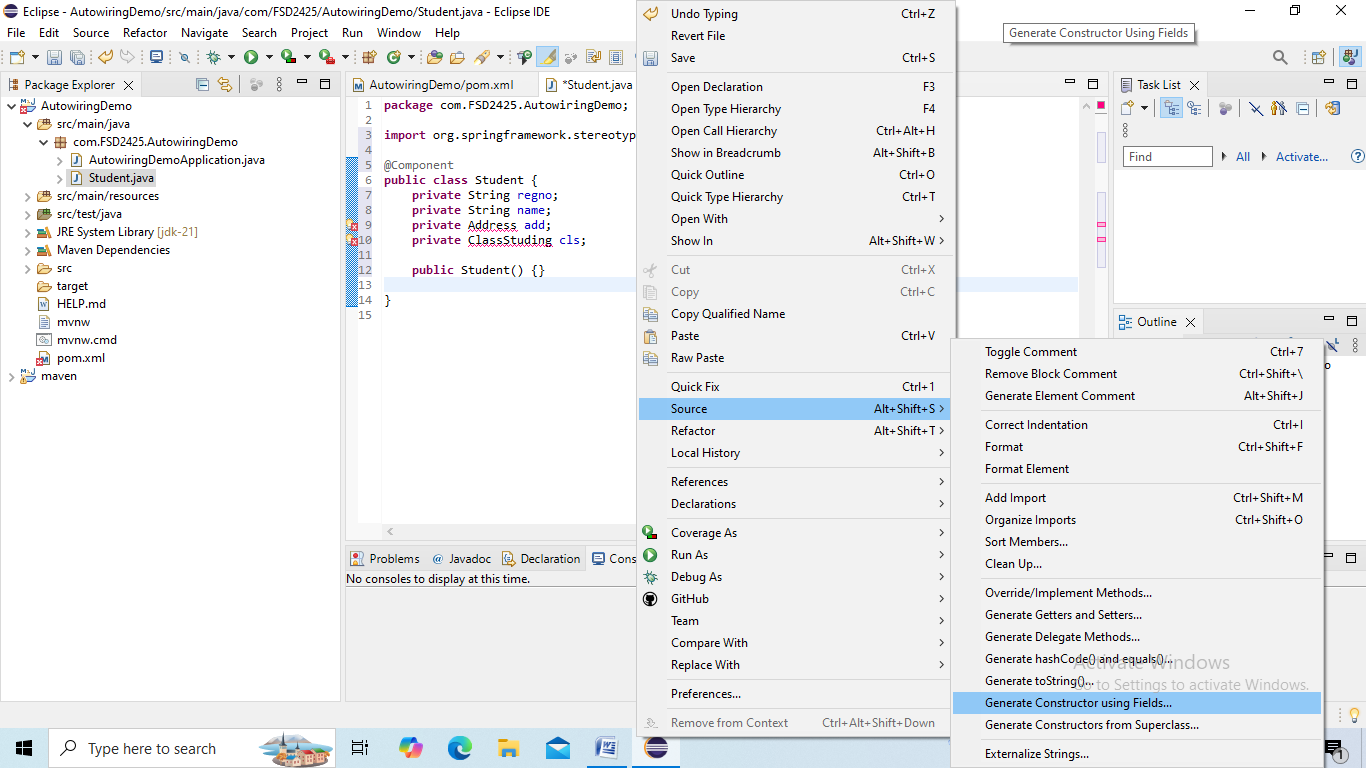
Step4: To create Student class:

Right click to your project🡪New🡪Class🡪Give Name:Student.

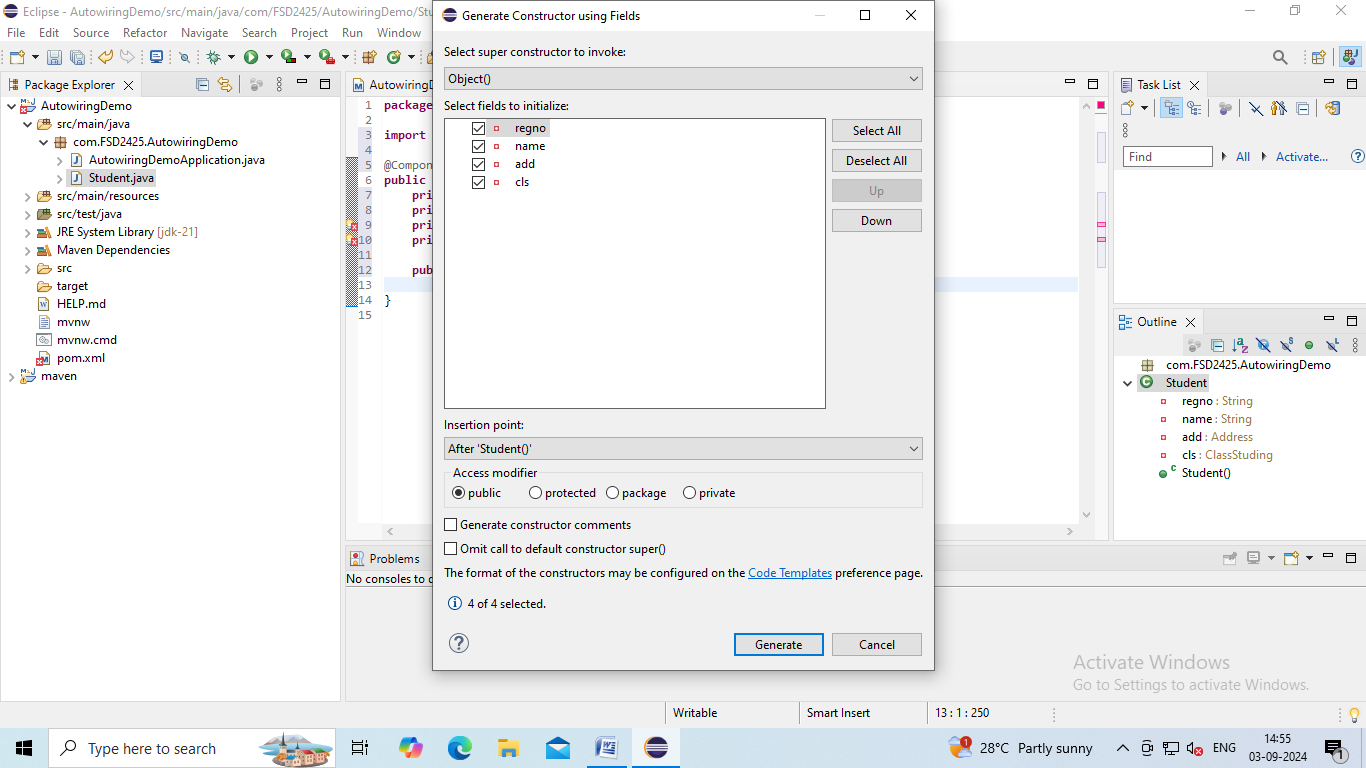


Step5: Write the code

To add constructor with field : Right click to editor🡪Source🡪Generate Constructor with fields.

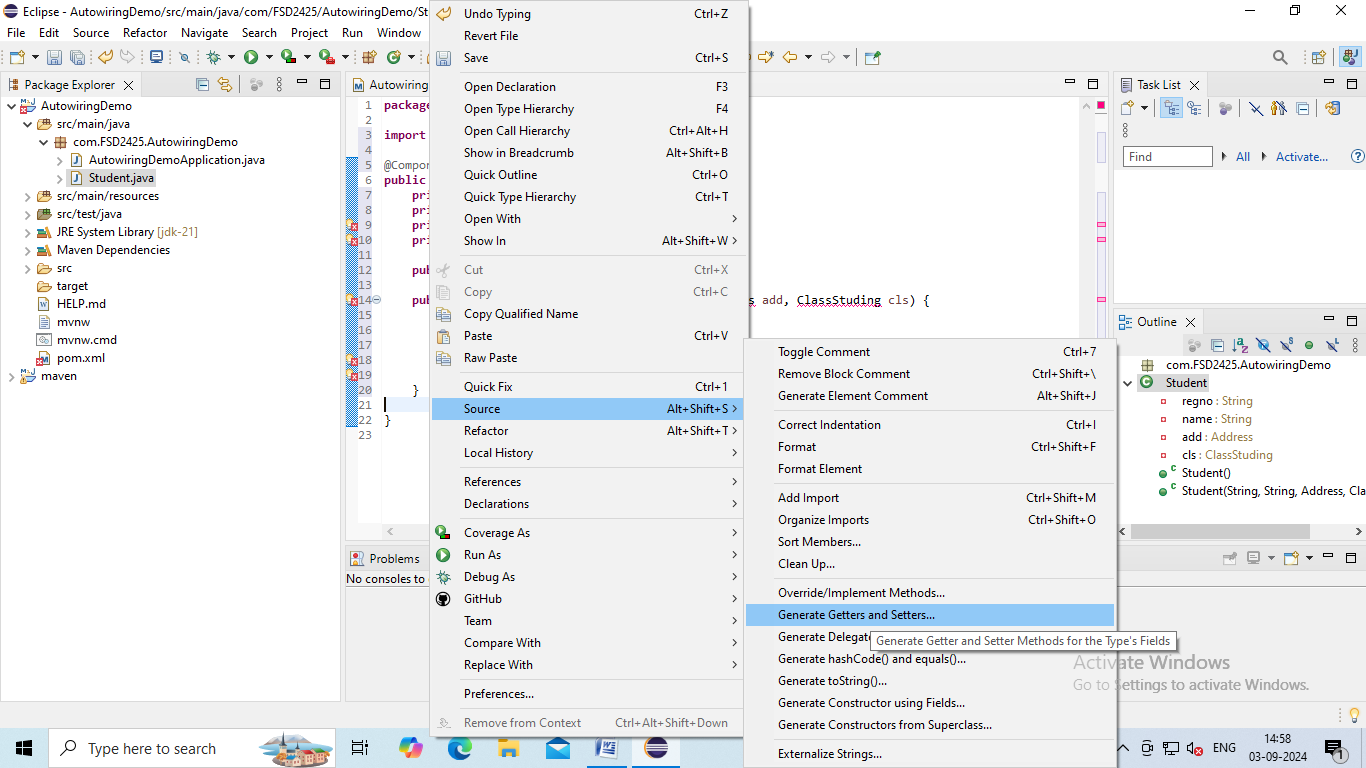


Step6: Click on Generate.

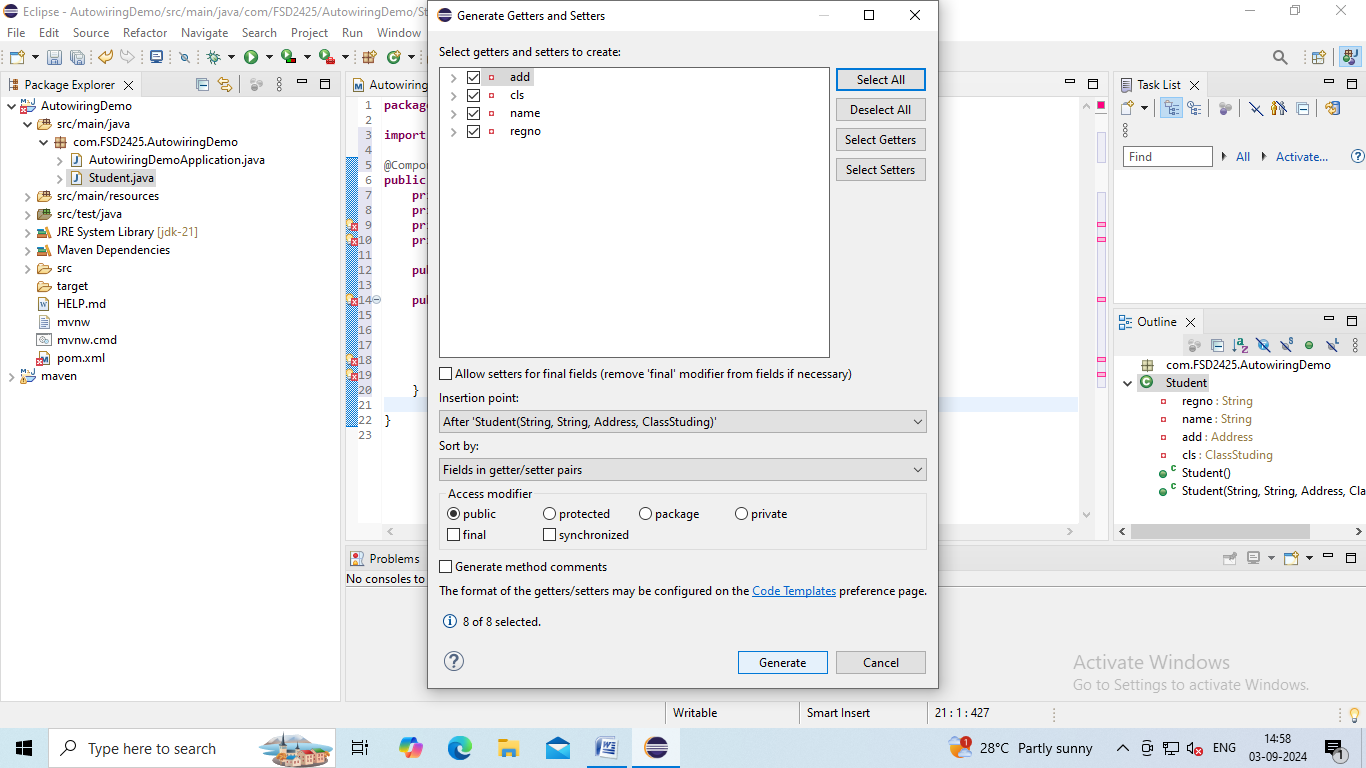


Step7: To add Getter and Setter methods.

Right click to your editor🡪Source🡪Generate Getters and Setters.

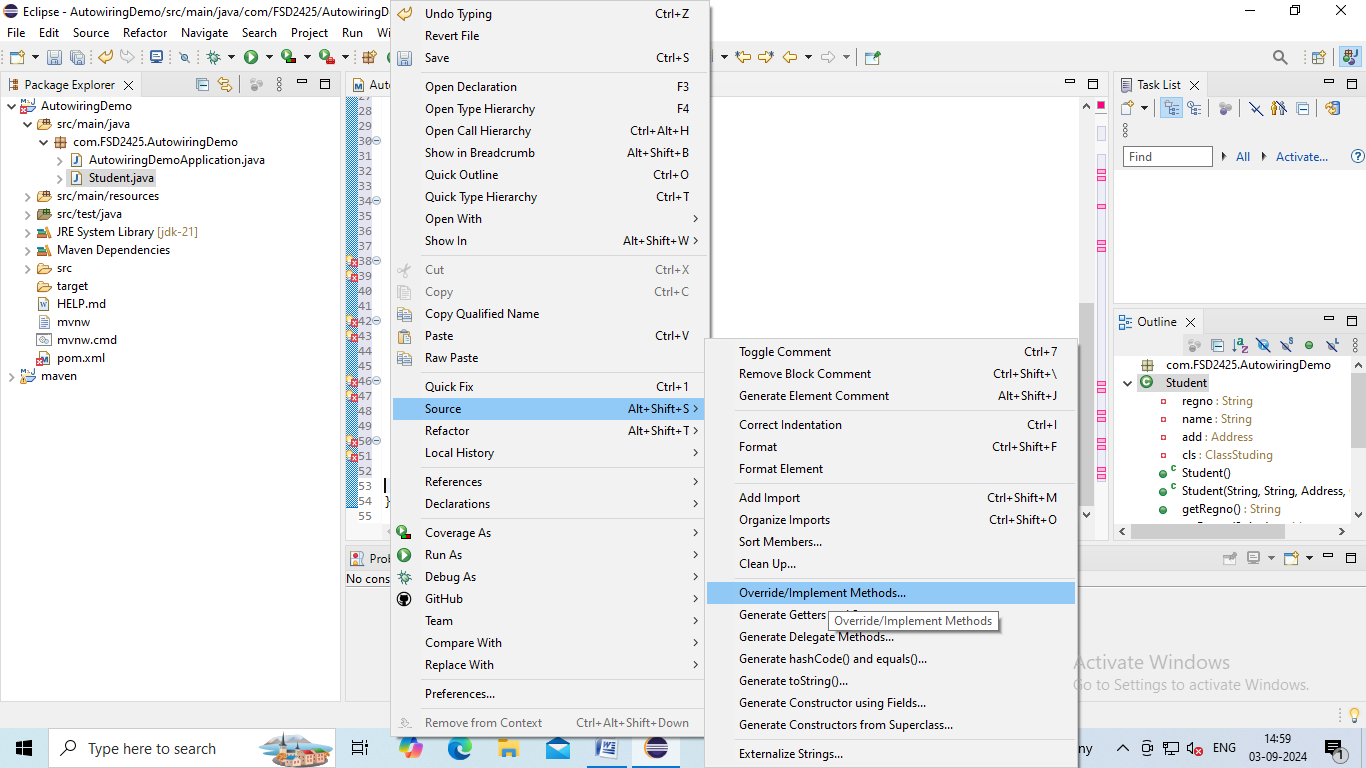


Step8: Click on Generate.

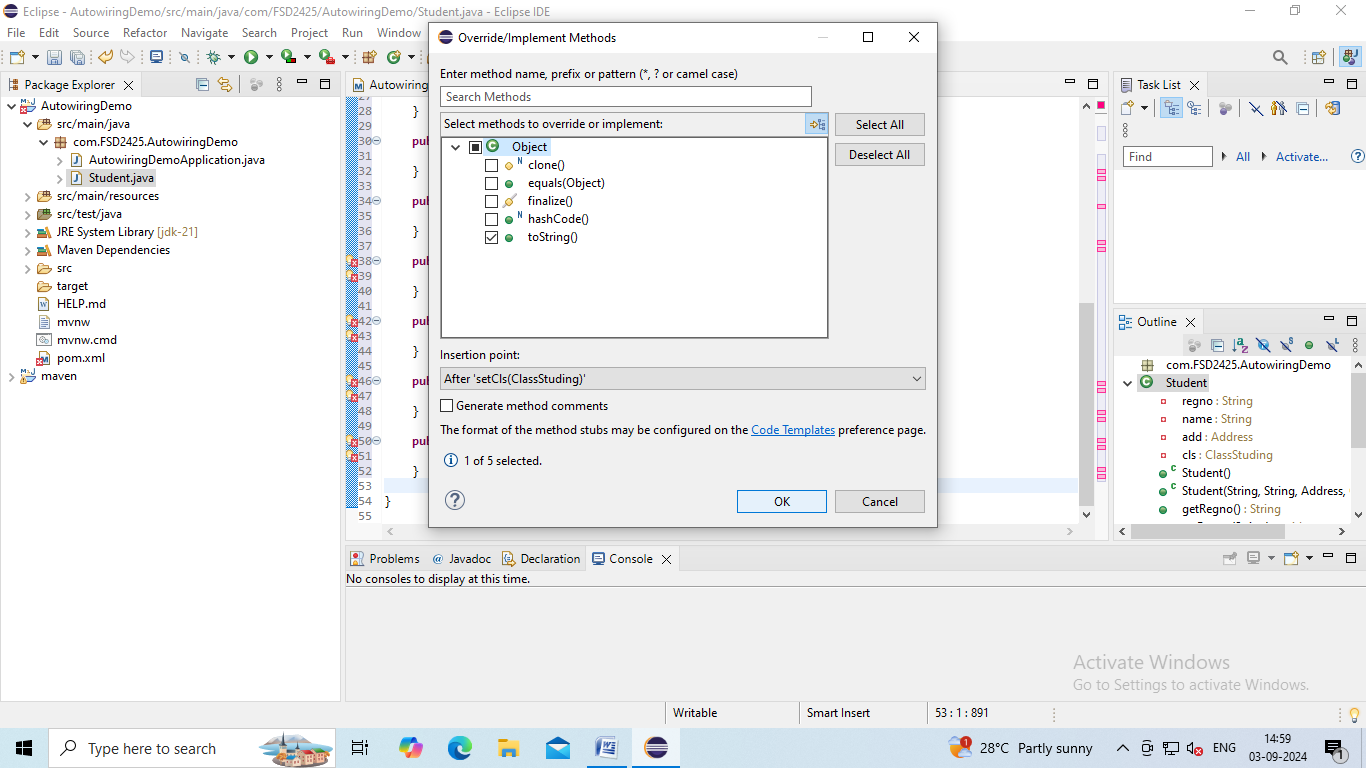


Step9: To override:

Right click to your editor🡪Source🡪Override/Implement methods.



Step10: Click on OK.



**Student.java:**

package com.FSD2425.AutowiringDemo;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Component;

@Component

public class Student {

private String regno;

private String name;

@Autowired

@Qualifier("add2")

private Address add;

@Autowired

private ClassStuding cls;

public Student() {}

public Student(String regno, String name) {

super();

this.regno = regno;

this.name = name;

}

public String getRegno() {

return regno;

}

public void setRegno(String regno) {

this.regno = regno;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Address getAdd() {

return add;

}

public void setAdd(Address add) {

this.add = add;

}

public ClassStuding getCls() {

return cls;

}

public void setCls(ClassStuding cls) {

this.cls = cls;

}

@Override

public String toString() {

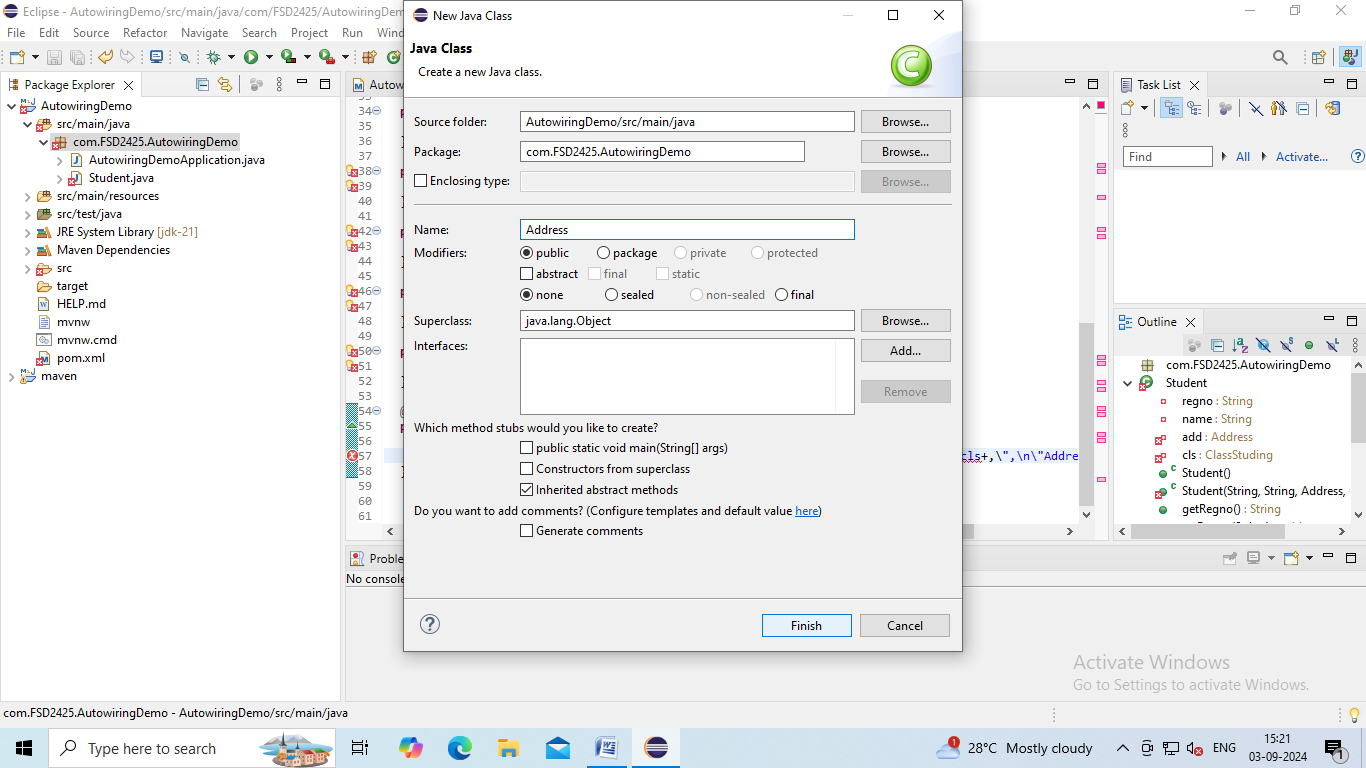
return("{\"RegNo\":\""+regno+"\",\n\"Name\":\""+name+"\",\n\"Class\":\""+cls+"\",\n\"Address\":\""+add+"\"}");

}

}

Step11: To create Address class:

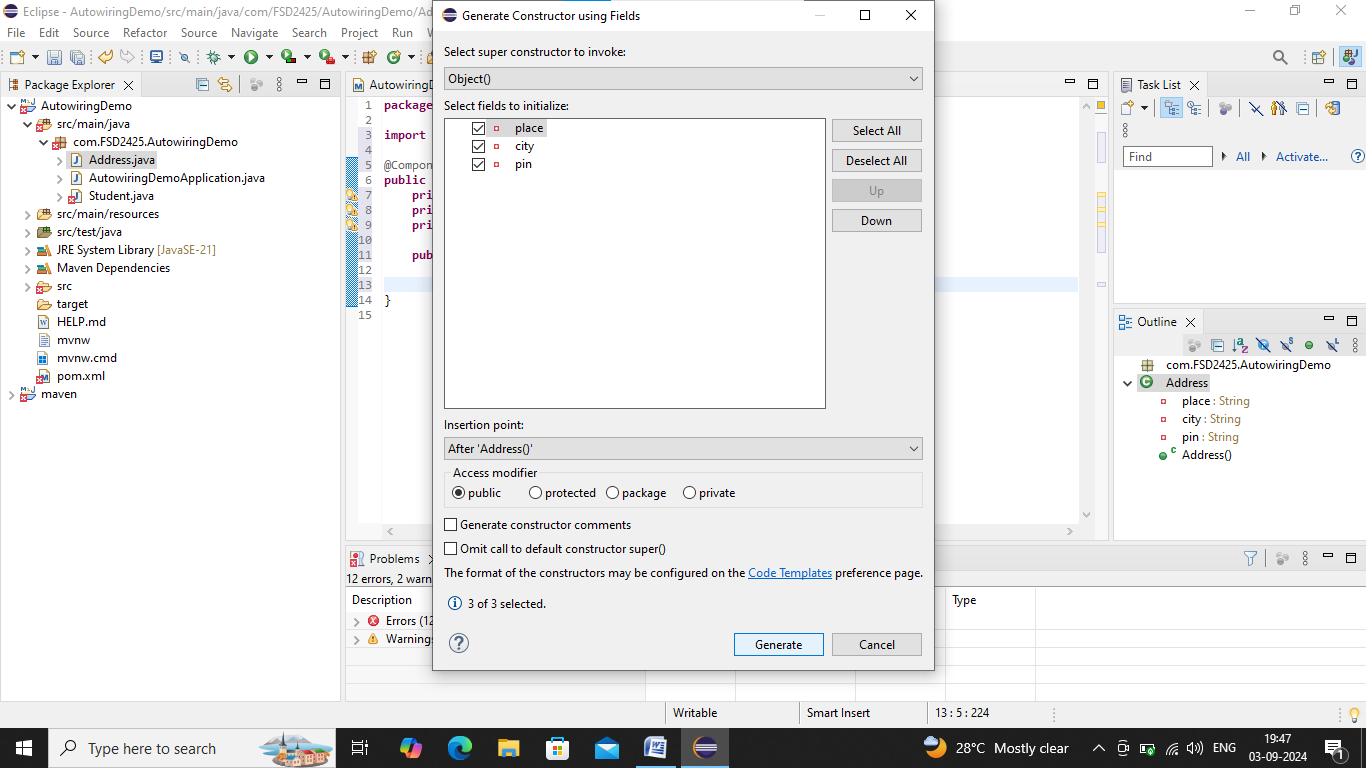
Right click to your project🡪New🡪Class🡪Give Name:Address🡪Click on Finish.



Step12: Write the code

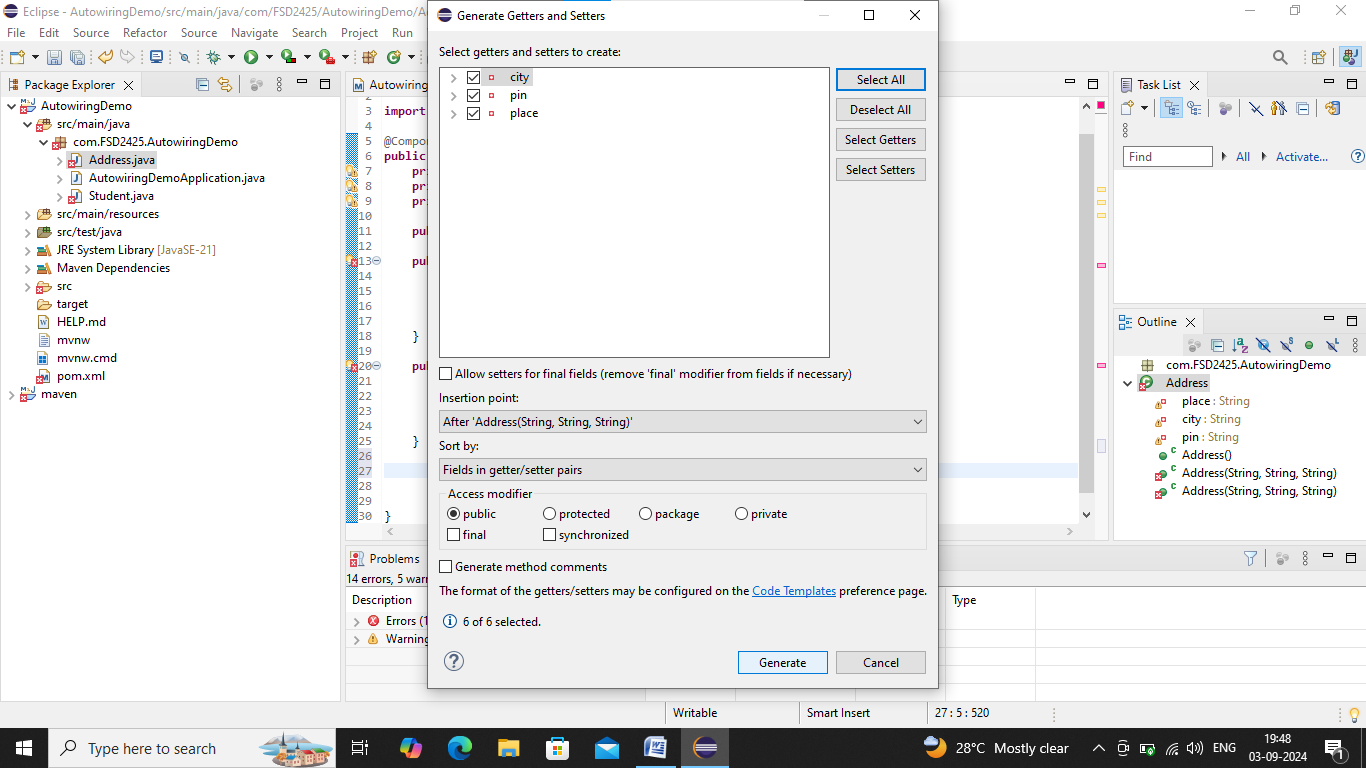
To add constructor with field : Right click to editor🡪Source🡪Generate Constructor with fields🡪Click on Generate.

Again Right click to editor🡪Source🡪Generate Constructor with fields🡪Deselect Pin🡪Click on Generate.



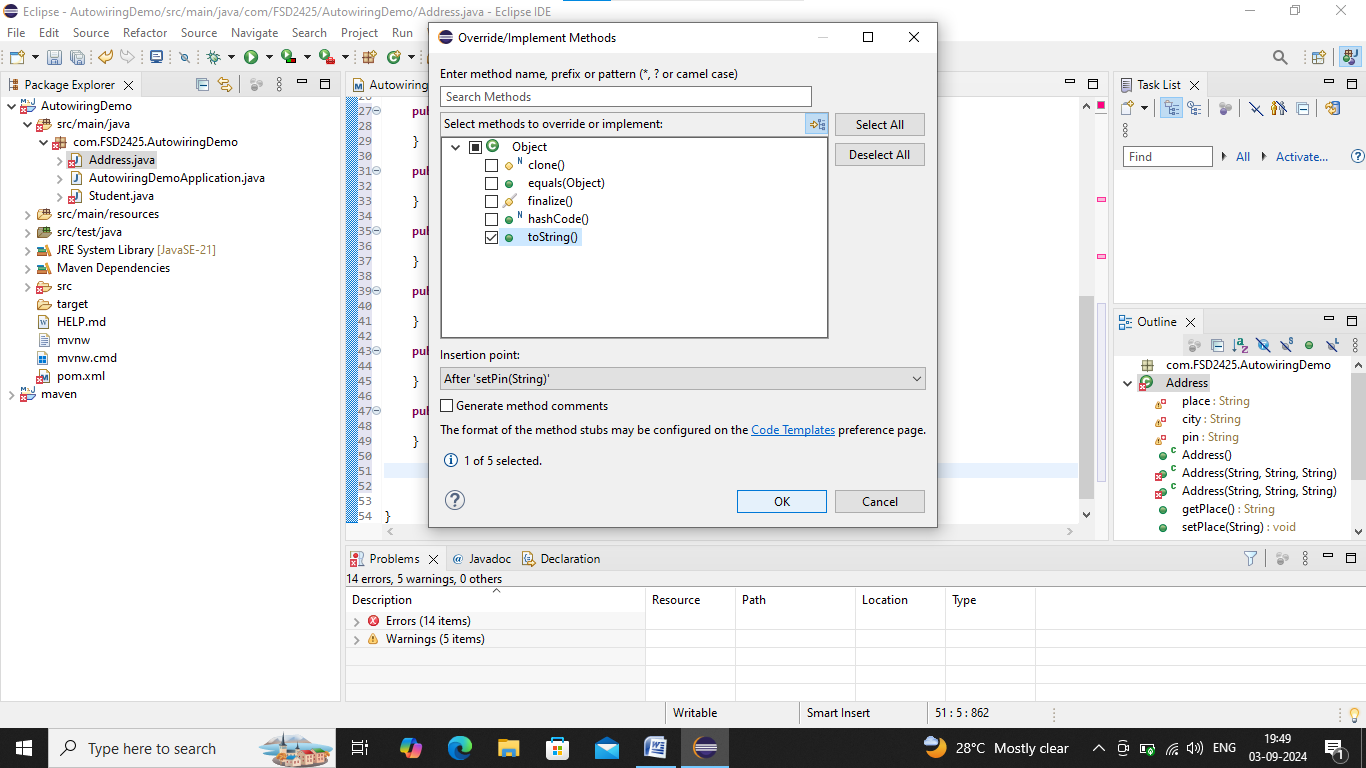
Step13: To add Getter and Setter methods.

Right click to your editor🡪Source🡪Generate Getters and Setters🡪Click on Generate.



Step14: To override:

Right click to your editor🡪Source🡪Override/Implement methods🡪Select toString🡪Click on ok.



**Address.java:**

**package** com.FSD2425.AutowiringDemo;

**import** org.springframework.stereotype.Component;

@Component

**public** **class** Address {

**private** String place;

**private** String city;

**private** String pin;

**public** Address() {}

**public** Address(String place, String city, String pin) {

**super**();

**this**.place = place;

**this**.city = city;

**this**.pin = pin;

}

**public** Address(String place, String city) {

**super**();

**this**.place = place;

**this**.city = city;

**this**.pin ="Pincode not defined";

}

**public** String getPlace() {

**return** place;

}

**public** **void** setPlace(String place) {

**this**.place = place;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getPin() {

**return** pin;

}

**public** **void** setPin(String pin) {

**this**.pin = pin;

}

@Override

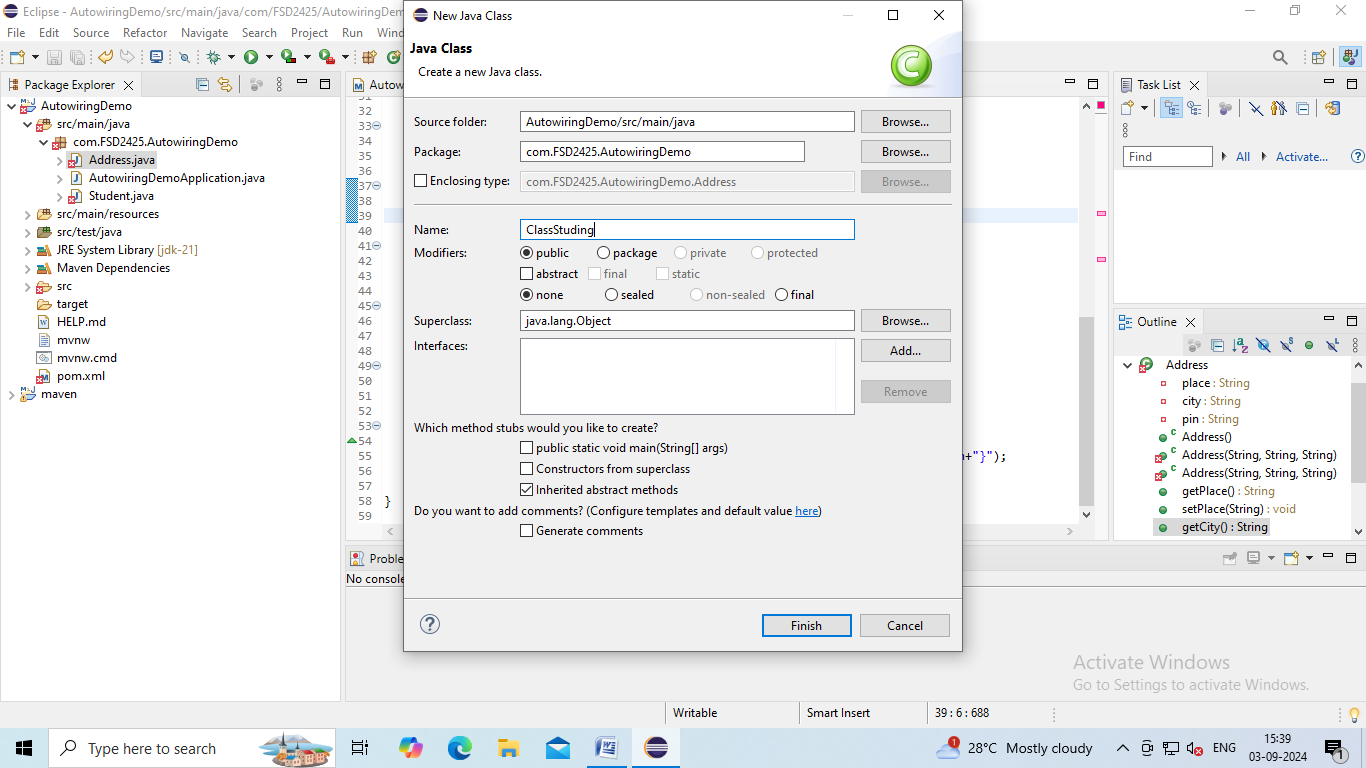
**public** String toString() {

**return** ("{\"Place\":\""+place+"\",\n\"City\":\""+city+"\",\n\"Pin\":\""+pin+"}");

}

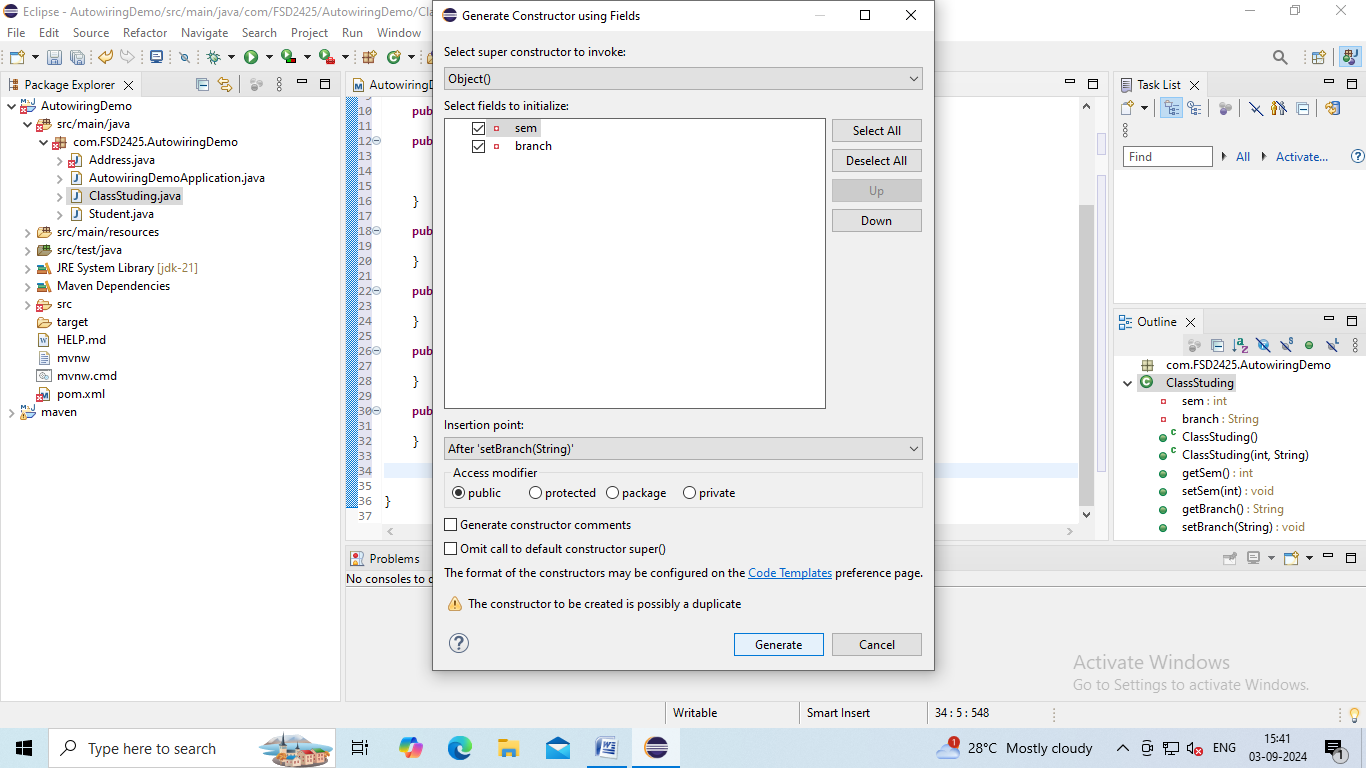
}

Step15: To create ClassStuding class:

Right click to your project🡪New🡪Class🡪Give Name:ClassStuding🡪Click on Finish.

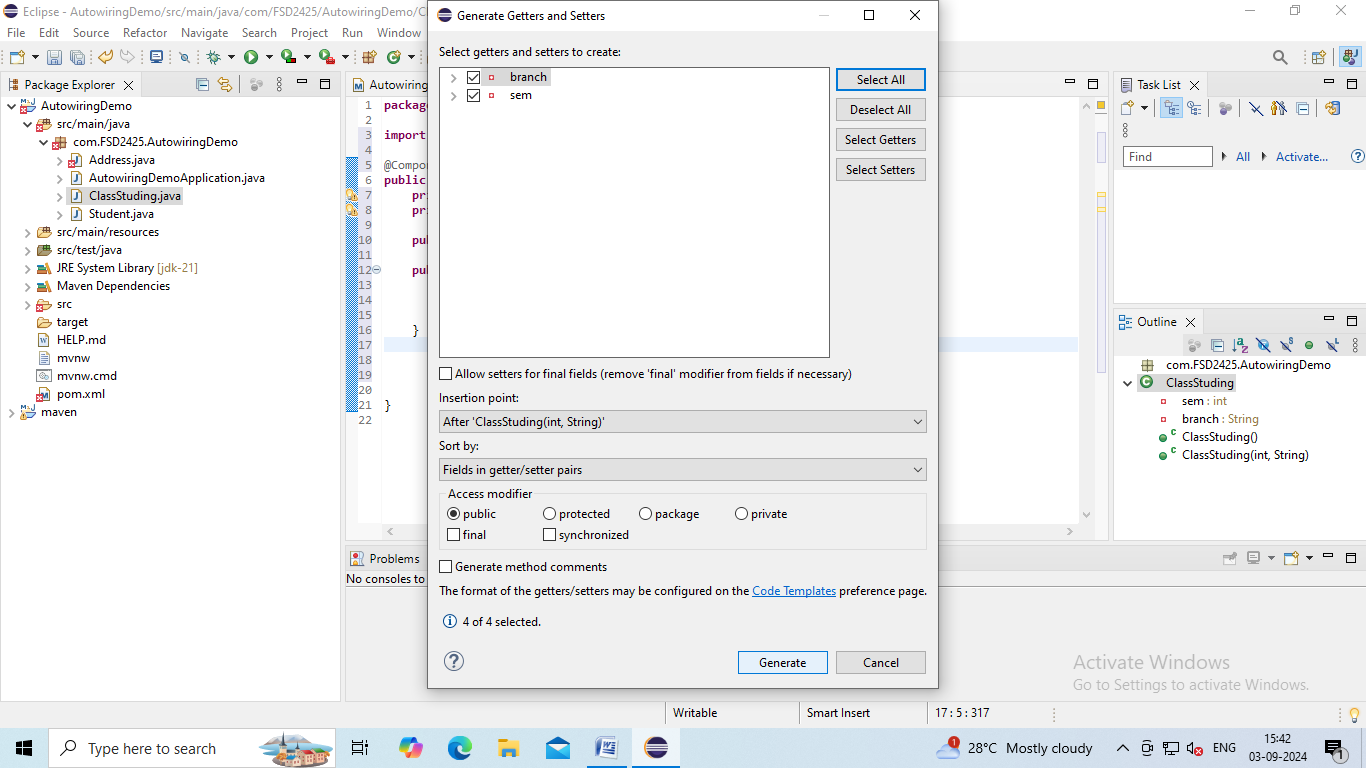
Step16: Write the code

To add constructor with field : Right click to editor🡪Source🡪Generate Constructor with fields🡪Click on Generate.



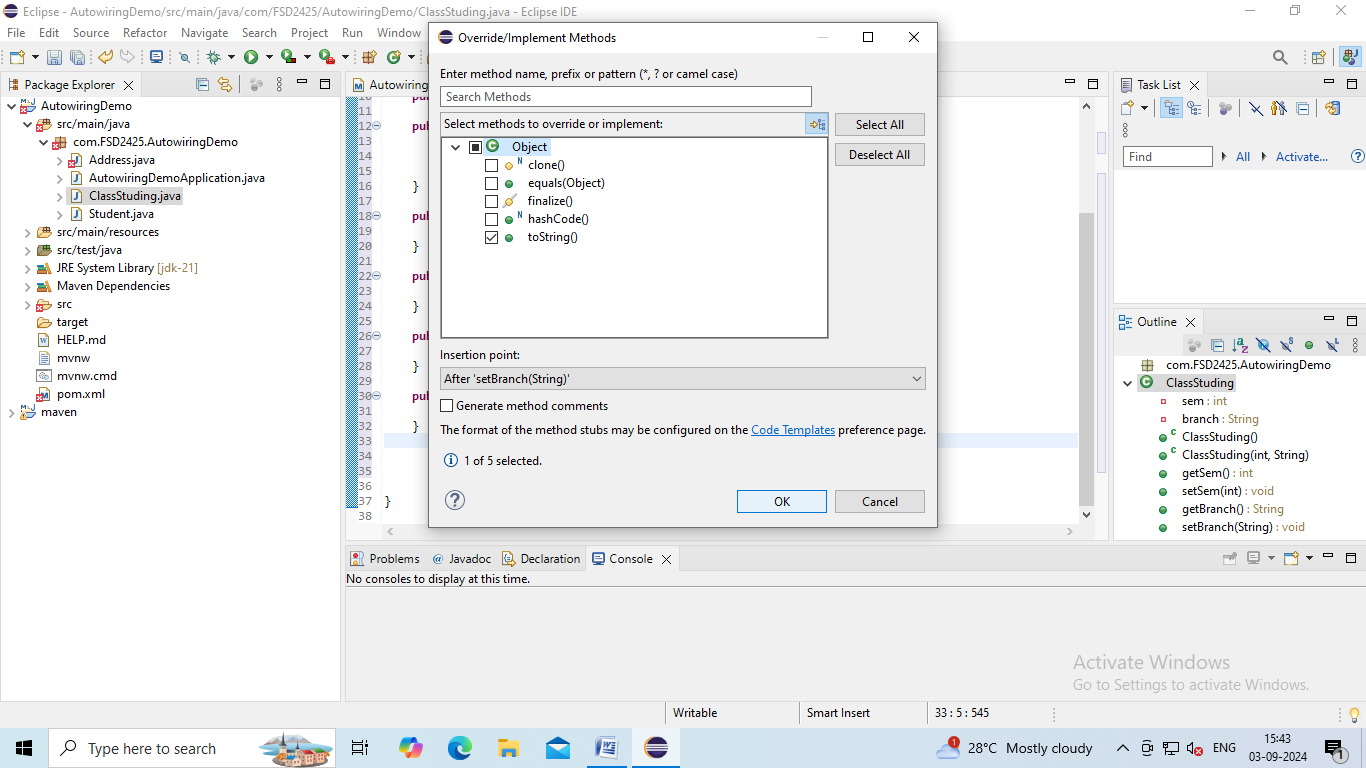
Step17: To add Getter and Setter methods.

Right click to your editor🡪Source🡪Generate Getters and Setters🡪Click on Generate.



Step18: To override:

Right click to your editor🡪Source🡪Override/Implement methods🡪Select toString🡪Click on ok.



**ClassStuding.java:**

**package** com.FSD2425.AutowiringDemo;

**import** org.springframework.stereotype.Component;

@Component

**public** **class** ClassStuding {

**private** **int** sem;

**private** String branch;

**public** ClassStuding() {}

**public** ClassStuding(**int** sem, String branch) {

**super**();

**this**.sem = sem;

**this**.branch = branch;

}

**public** **int** getSem() {

**return** sem;

}

**public** **void** setSem(**int** sem) {

**this**.sem = sem;

}

**public** String getBranch() {

**return** branch;

}

**public** **void** setBranch(String branch) {

**this**.branch = branch;

}

@Override

**public** String toString() {

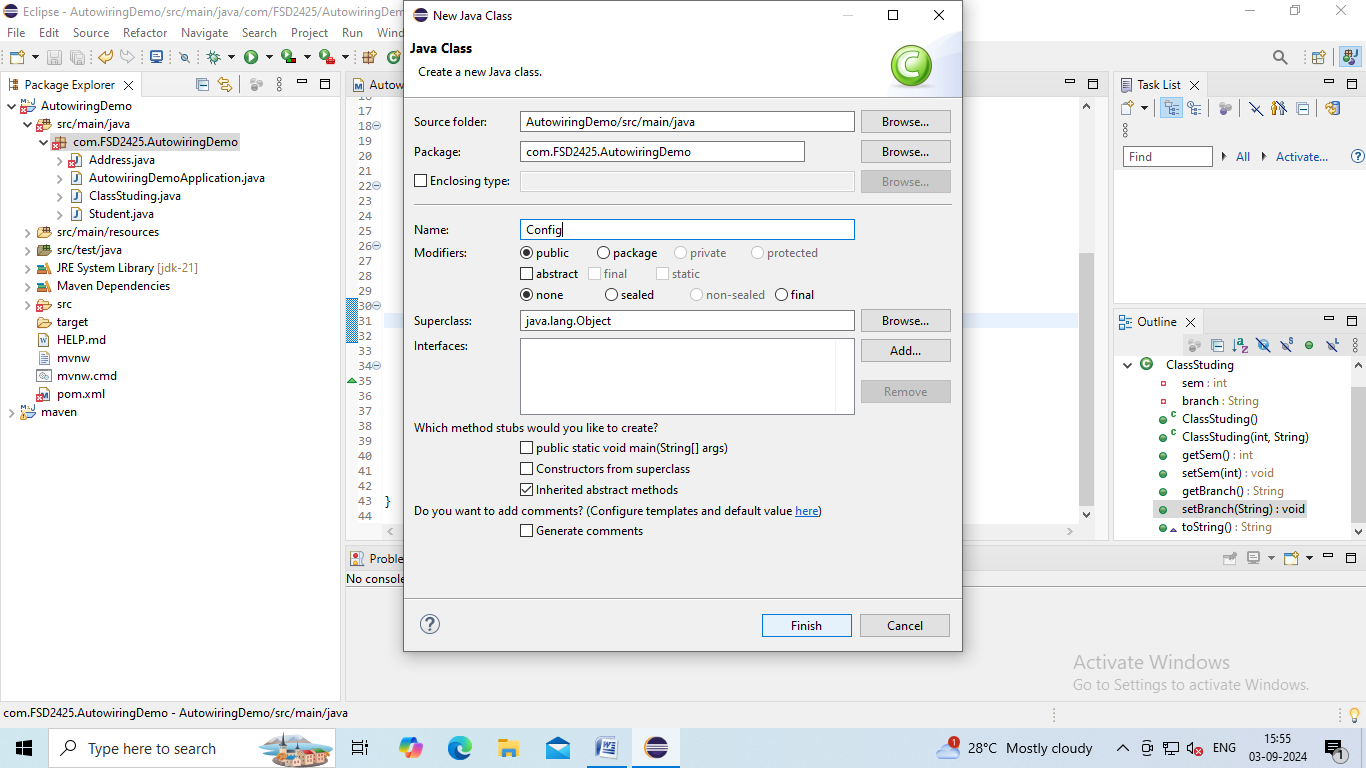
**return** ("{\"Semester\":\""+sem+"\",\n\"Branch\":\""+branch+"}");

}

}

Step19: To create Config class:

Right click to your project🡪New🡪Class🡪Give Name:Config🡪Click on Finish.



Step20: Write code as follows:

**Config.js:**

package com.FSD2425.AutowiringDemo;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.Primary;

import org.springframework.context.annotation.Scope;

@Configuration

public class Config {

@Bean(name="add1")

public Address add1 () {

return new Address("Kundadka","Viitla", "574243");

}

@Bean(name="add2")

public Address add2 () {

return new Address("Odiyoor","Kanyana", "574240");

}

@Bean (name="add3")

public Address add3 () {

return new Address("Adoor","Kasaragod");

}

@Bean

@Primary

public ClassStuding clstd() {

return new ClassStuding(5,"CS");

}

@Bean

@Scope("singleton")

public Student student() {

return new Student("Varshita","391cs22054");

}

}

Step21: Write the code **in AutowiringDemoApplication.java:**

package com.FSD2425.AutowiringDemo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ConfigurableApplicationContext;

@SpringBootApplication

public class AutowiringDemoApplication {

public static void main(String[] args) {

ConfigurableApplicationContext cn = SpringApplication.run(AutowiringDemoApplication.class, args);

Student s1 = cn.getBean(Student.class);

System.out.println("Student s1:");

System.out.println(s1);

System.out.println("Hash code: " + s1.hashCode());

Student s2 = cn.getBean(Student.class);

System.out.println("Student s2:");

System.out.println(s2);

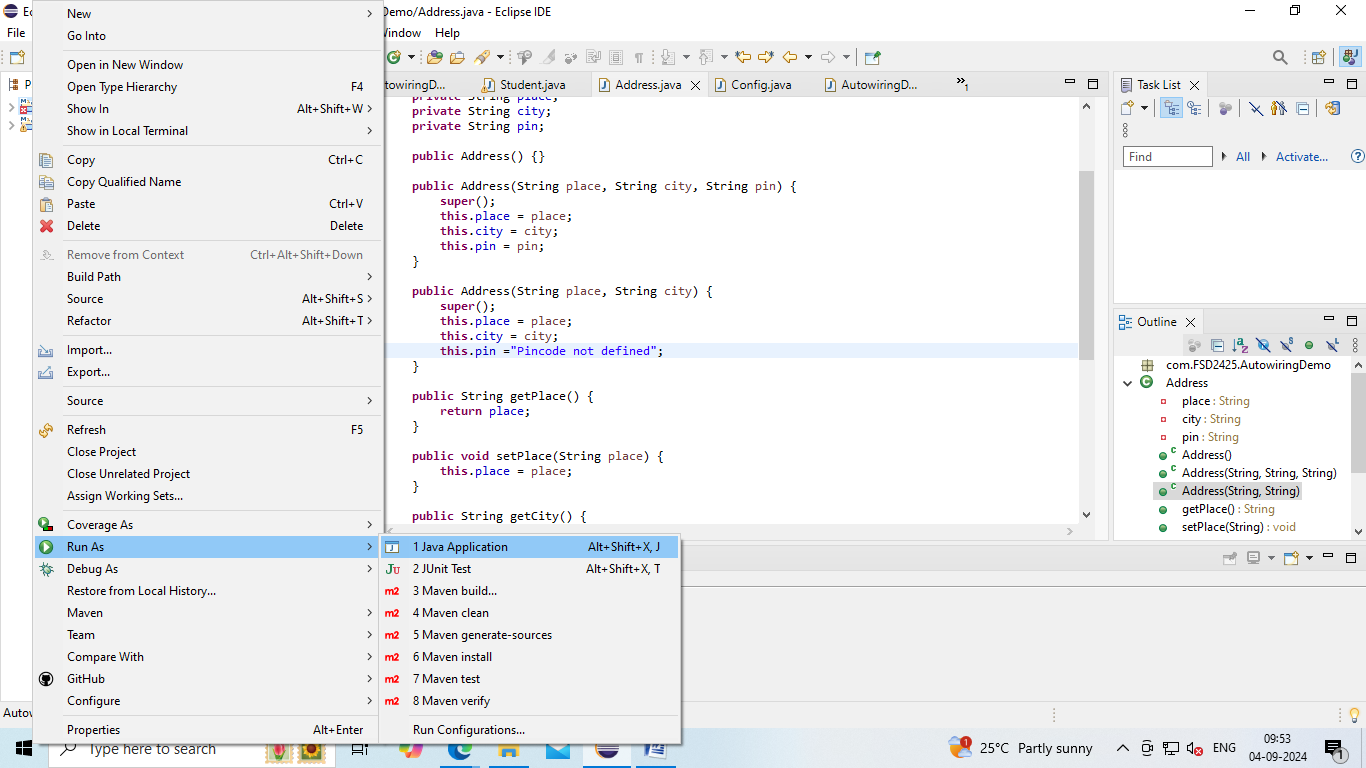
System.out.println("Hash code: " + s2.hashCode());

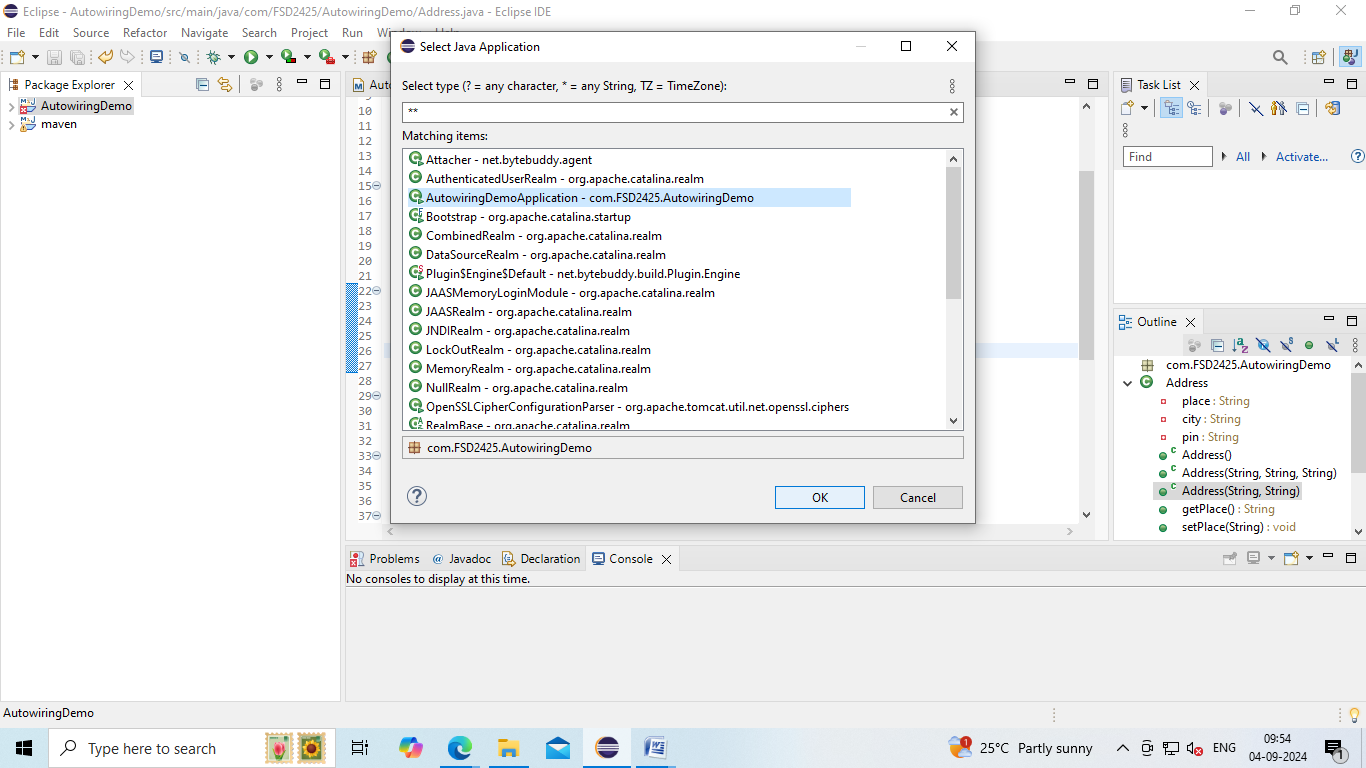
}

}

Step22: Now to run the code:

Right click to your project🡪Run as🡪Java Application🡪Search your project🡪Click on OK.





**Outputs:**

**O/P1:**

Student s1:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Kundadka",

"City":"Viitla",

"Pin":"574243}"}

Hash code: 45639856

Student s2:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Kundadka",

"City":"Viitla",

"Pin":"574243}"}

Hash code: 1457833923

**O/P2:**

Do change as @Qualifier("add2") in Student.java

Student s1:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Odiyoor",

"City":"Kanyana",

"Pin":"574240}"}

Hash code: 1514247686

Student s2:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Odiyoor",

"City":"Kanyana",

"Pin":"574240}"}

Hash code: 777724086

**O/P3**:

Do changes as @Qualifier("add3") in Student.java

Student s1:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Adoor",

"City":"Kasaragod",

"Pin":"Pincode not defined}"}

Hash code: 2145036079

Student s2:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Adoor",

"City":"Kasaragod",

"Pin":"Pincode not defined}"}

Hash code: 908085418

**O/P4:**

Do changes in

@Scope("singleton") in Config.java and

@Qualifier("add3") in Student.java

Student s1:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Adoor",

"City":"Kasaragod",

"Pin":"Pincode not defined}"}

Hash code: 1356622380

Student s2:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Adoor",

"City":"Kasaragod",

"Pin":"Pincode not defined}"}

Hash code: 1356622380

**O/P5**: Do changes as @Qualifier("add3") in Student.java

Student s1:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Odiyoor",

"City":"Kanyana",

"Pin":"574240}"}

Hash code: 280957370

Student s2:

{"RegNo":"Varshita",

"Name":"391cs22054",

"Class":"{"Semester":"5",

"Branch":"CS}",

"Address":"{"Place":"Odiyoor",

"City":"Kanyana",

"Pin":"574240}"}

Hash code: 280957370