*//Ques1: Write a program to demonstrate Tightly Coupled code.*

**package** springFrameworkExercise;

**public class** Ques1 {

**public static void** main(String[] args) {

Manager manager = **new** Manager();

manager.func();

}

}

**class** Manager{

Worker **worker** = **new** LazyWorker();

**public void** func(){

**worker**.doWork();

}

}

**interface** Worker{

**public void** doWork();

}

**class** LazyWorker **implements** Worker{

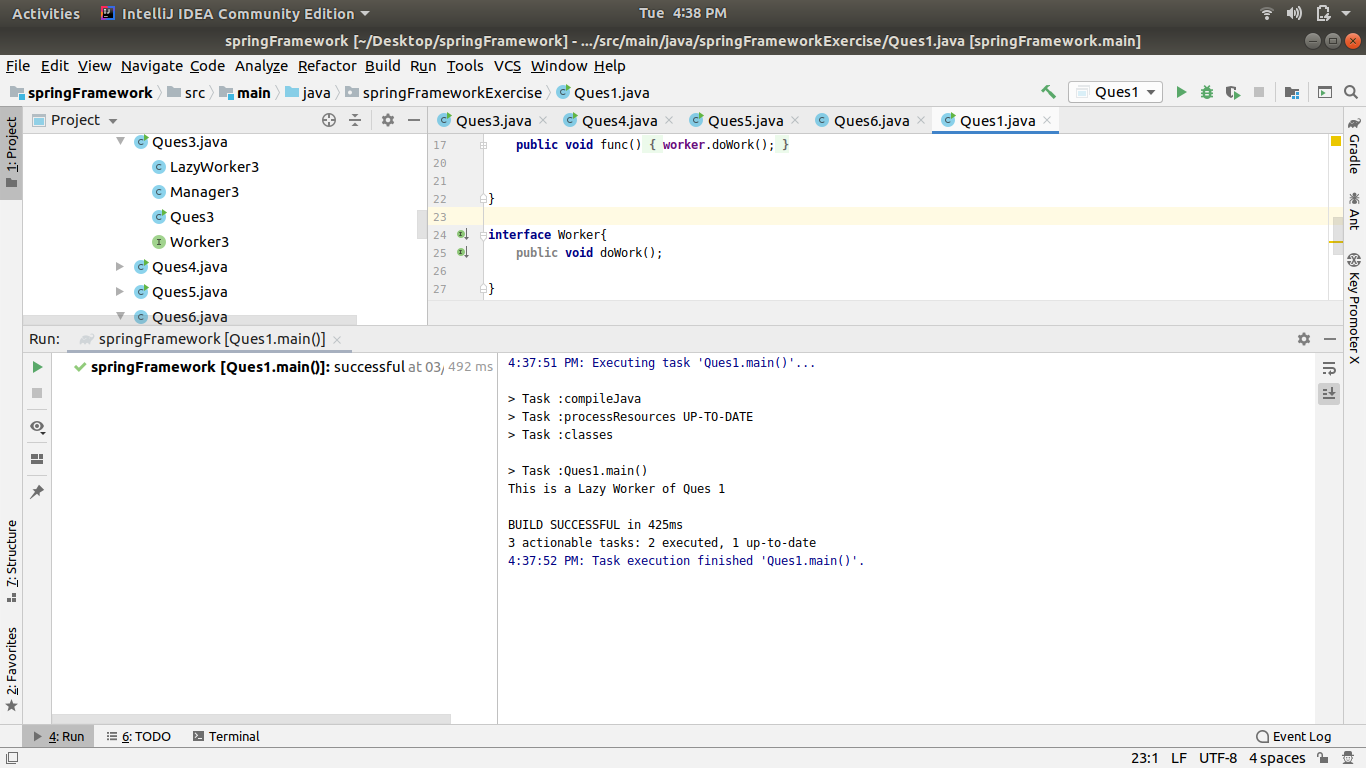
@Override

**public void** doWork() {

System.***out***.println(**"This is a Lazy Worker"**);

}

}



*//Ques2: Write a program to demonstrate Loosely Coupled code.*

**package** springFrameworkExercise;

**public class** Ques2 {

**public static void** main(String[] args) {

**new** Manager2(**new** LazyWorker2()).func();

*// manager2.func();*

}

}

**class** Manager2{

Worker2 **worker2**;

Manager2(Worker2 worker2){

**this**.**worker2** = worker2;

*// func();*

}

**public void** func(){

**worker2**.doWork();

}

}

**interface** Worker2{

**public void** doWork();

}

**class** LazyWorker2 **implements** Worker2{

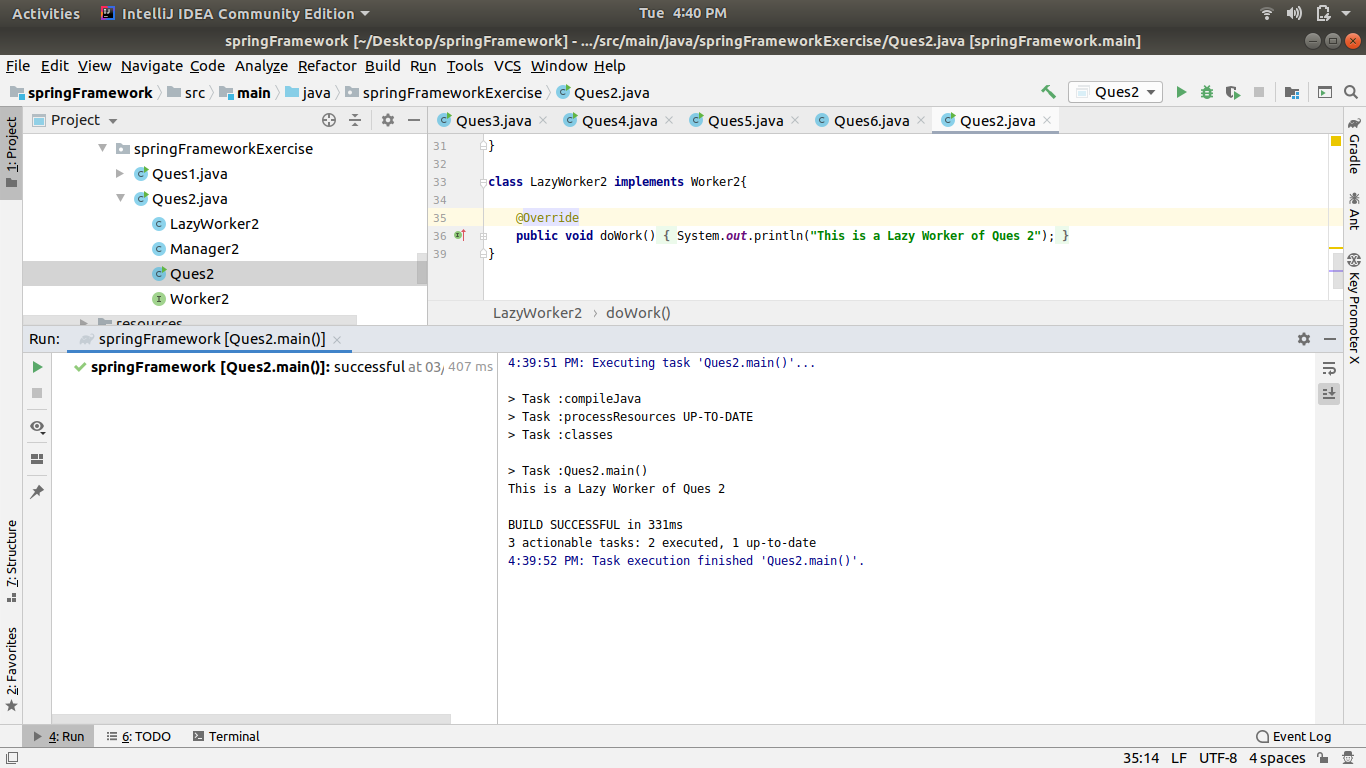
@Override

**public void** doWork() {

System.***out***.println(**"This is a Lazy Worker of Ques 2"**);

}

}



*//Ques3: Use @Compenent and @Autowired annotations to in Loosely Coupled code for dependency management*

**package** com.tothenew.springbootcamp.springFramework;

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

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.ApplicationContext;

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

@SpringBootApplication

**public class** Ques3 {

**public static void** main(String[] args) {

ApplicationContext ac3 = SpringApplication.*run*(Ques3.**class**, args);

Manager3 manager3 = ac3

.getBean(Manager3.**class**);

manager3.abc();

}

}

@Component

**class** Manager3{

@Autowired

Worker3 **worker3**;

**public void** abc(){

**worker3**.doWork();

}

}

**interface** Worker3{

**void** doWork();

}

@Component

**class** LazyWorker3 **implements** Worker3{

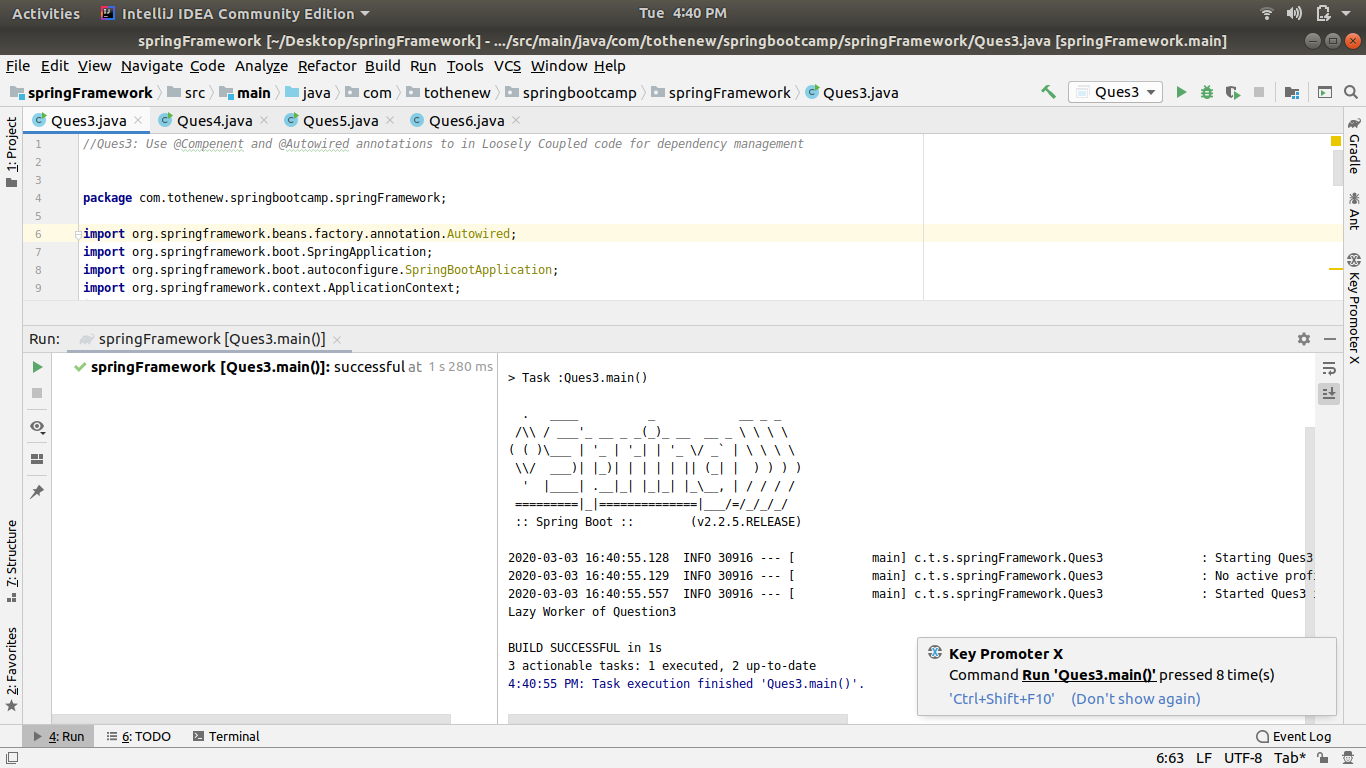
@Override

**public void** doWork() {

System.***out***.println(**"Lazy Worker of Question3"**);

}

}



*//Ques 4: Get a Spring Bean from application context and display its properties.*

**package** com.tothenew.springbootcamp.springFramework;

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

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.context.ApplicationContext;

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

**public class** Ques4 {

**public static void** main(String[] args) {

ApplicationContext ac4 = SpringApplication.*run*(Ques3.**class**, args);

Manager4 manager4 = ac4.getBean(Manager4.**class**);

System.***out***.println(manager4.**property1**);

}

}

@Component

**class** Manager4{

**int property1** = 100;

@Autowired

Worker4 **worker4**;

**public void** abc(){

**worker4**.doWork();

}

}

**interface** Worker4{

**void** doWork();

}

@Component

**class** LazyWorker4 **implements** Worker4{

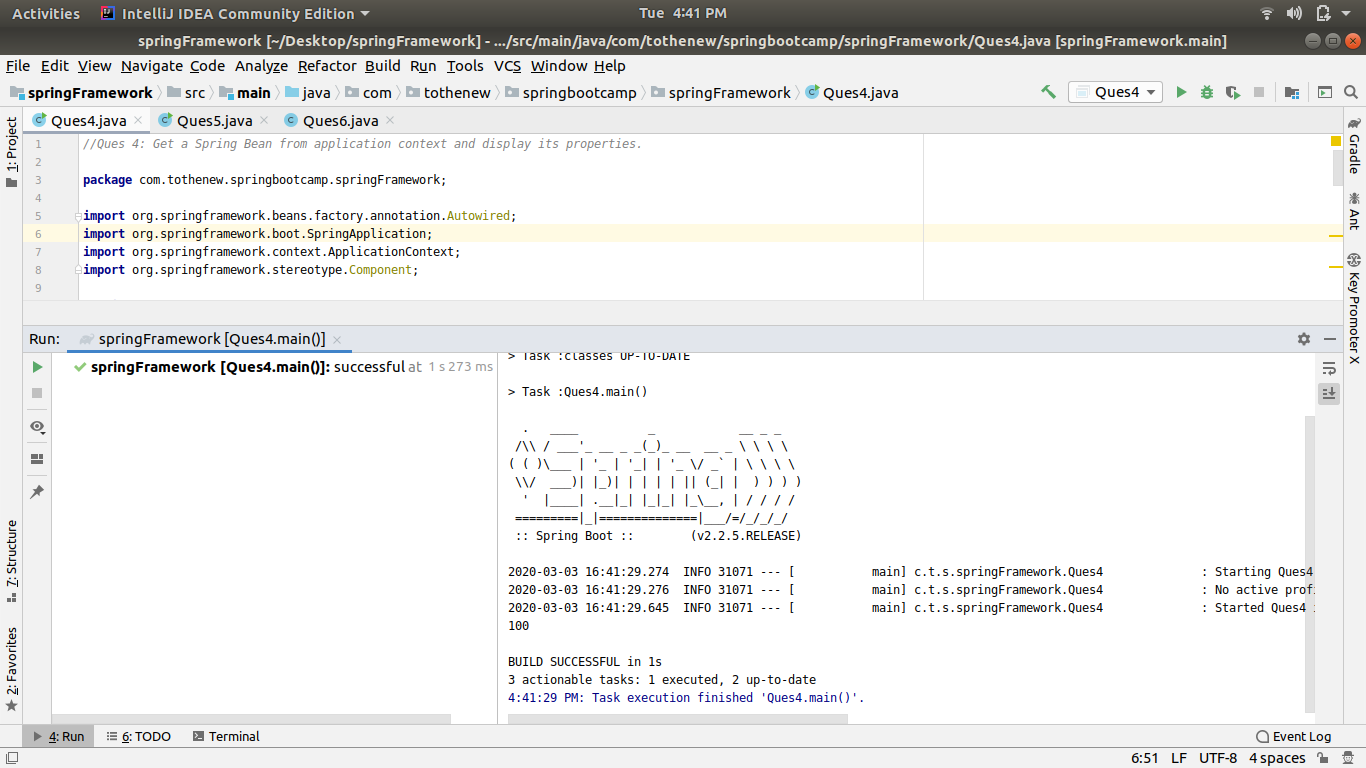
@Override

**public void** doWork() {

System.***out***.println(**"Lazy Worker of Question4"**);

}

}



*//Ques 5: Demonstrate how you will resolve ambiguity while autowiring bean (Hint : @Primary)*

**package** com.tothenew.springbootcamp.springFramework;

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

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.annotation.Primary;

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

@SpringBootApplication

**public class** Ques5 {

**public static void** main(String[] args) {

ApplicationContext ac5 = SpringApplication.*run*(Ques3.**class**, args);

Manager5 manager5 = ac5.getBean(Manager5.**class**);

manager5.callDoWork();

}

}

@Component

**class** Manager5{

Worker5 **worker5**;

@Autowired

Manager5(Worker5 worker5){

**this**.**worker5** = worker5;

}

**public void** callDoWork(){

**worker5**.doWork();

}

}

**interface** Worker5{

**void** doWork();

}

@Component

@Primary

**class** LazyWorker5 **implements** Worker5{

@Override

**public void** doWork() {

System.***out***.println(**"Lazy Worker of Question 5"**);

}

}

@Component

**class** ExcellentWorker5 **implements** Worker5{

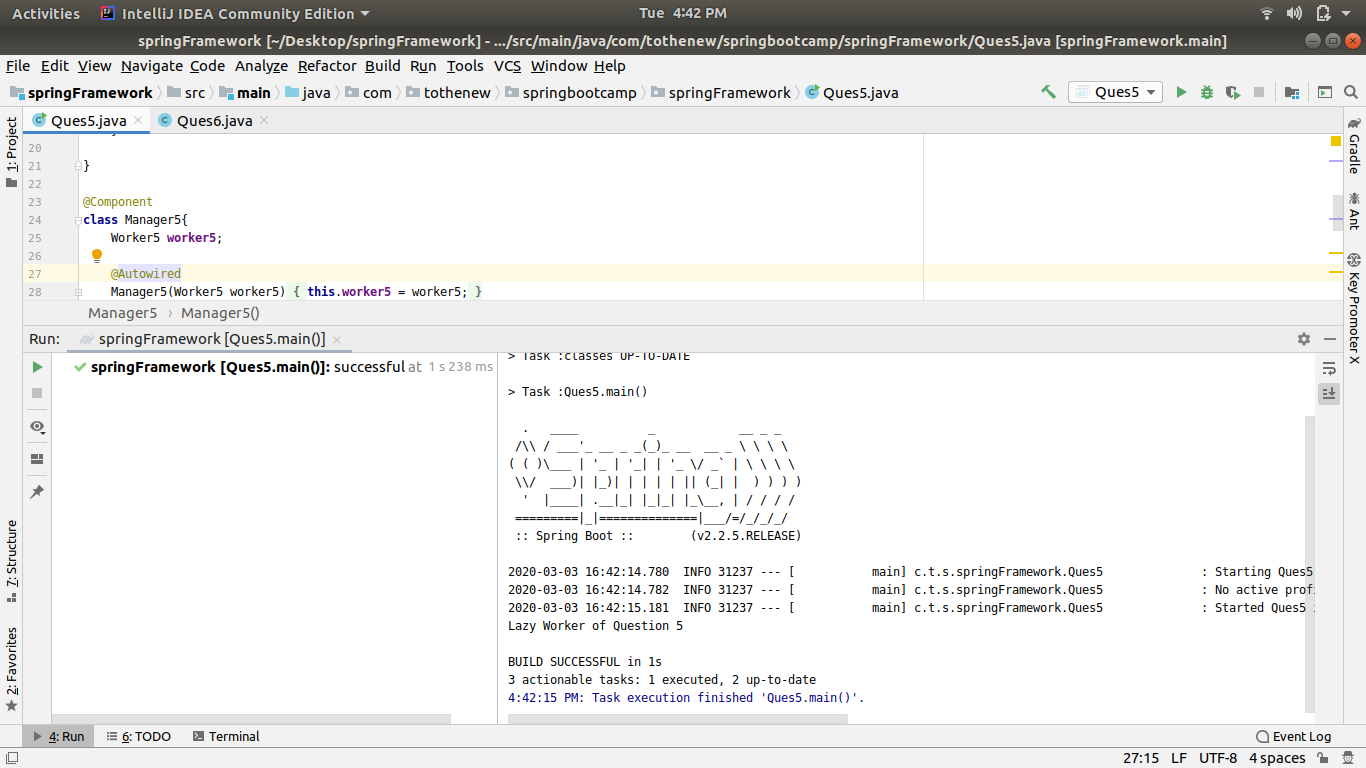
@Override

**public void** doWork() {

System.***out***.println(**"Excellent Worker of Question 5"**);

}

}



*//Ques 6:Perform Constructor Injection in a Spring Bean*

**package** com.tothenew.springbootcamp.springFramework;

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

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.annotation.Primary;

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

@SpringBootApplication

**public class** Ques6 {

**public static void** main(String[] args) {

ApplicationContext ac6 = SpringApplication.*run*(Ques3.**class**, args);

Manager6 manager6 = ac6.getBean(Manager6.**class**);

manager6.callDoWork();

}

}

@Component

**class** Manager6 {

@Autowired

Worker6 **worker6**;

Manager6(Worker6 worker5){

**this**.**worker6** = worker5;

}

**public void** callDoWork(){

**worker6**.doWork();

}

}

**interface** Worker6 {

**void** doWork();

}

@Component

**class** LazyWorker6 **implements** Worker6 {

@Override

**public void** doWork() {

System.***out***.println(**"Lazy Worker of Question 6"**);

}

}