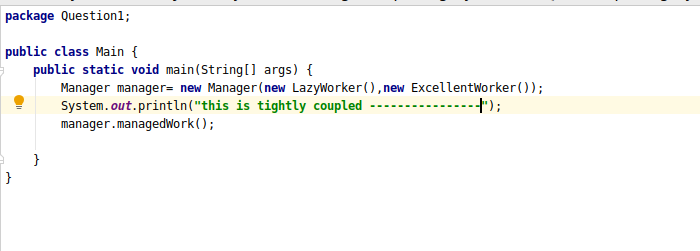
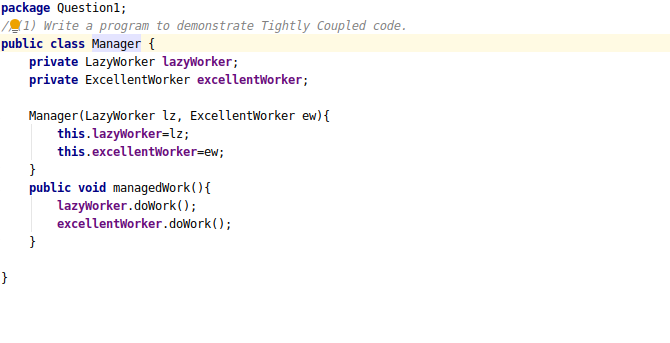
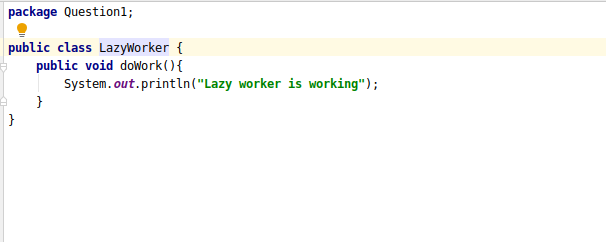
Exercise

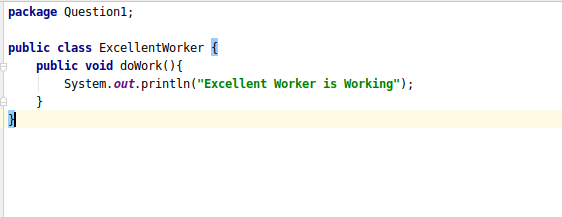
(1) Write a program to demonstrate Tightly Coupled code.

Ans:- There is shared memory. In tightly coupled multiprocessor system, data rate is high rather than loosely coupled multiprocessor system. In tightly coupled multiprocessor system, modules are connected through PMIN, IOPIN and ISIN networks.

SS-



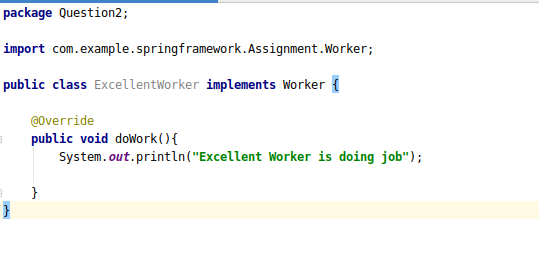


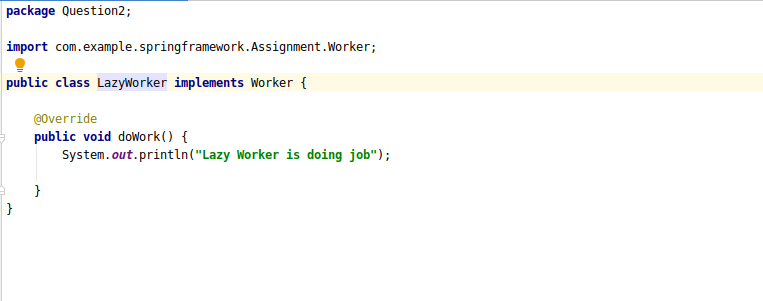


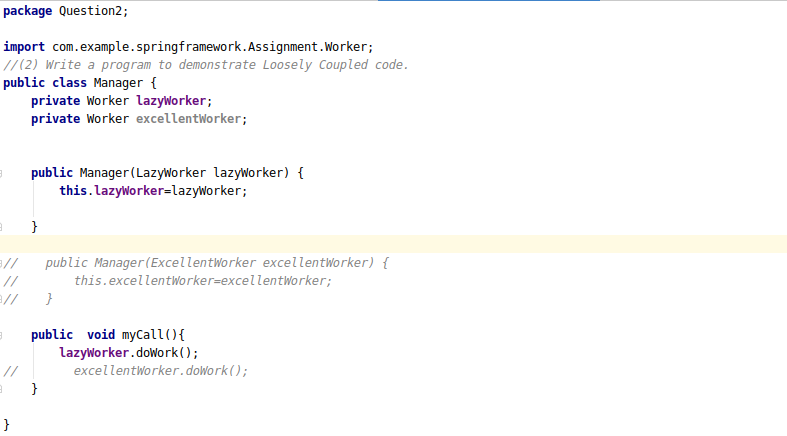
(2) Write a program to demonstrate Loosely Coupled code.

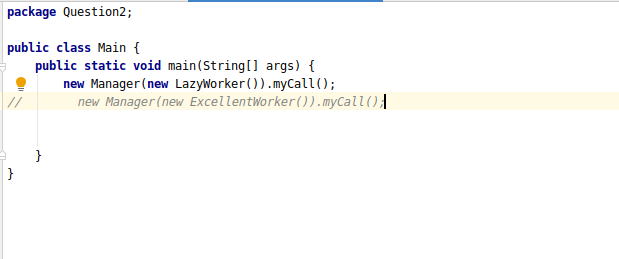
Ans:-There is distributed memory instead of shared memory. In loosely coupled multiprocessor system, data rate is low rather than tightly coupled multiprocessor system. In loosely coupled multiprocessor system, modules are connected through MTS (Message transfer system) network.

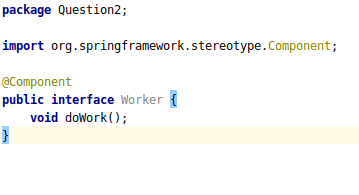
SS-







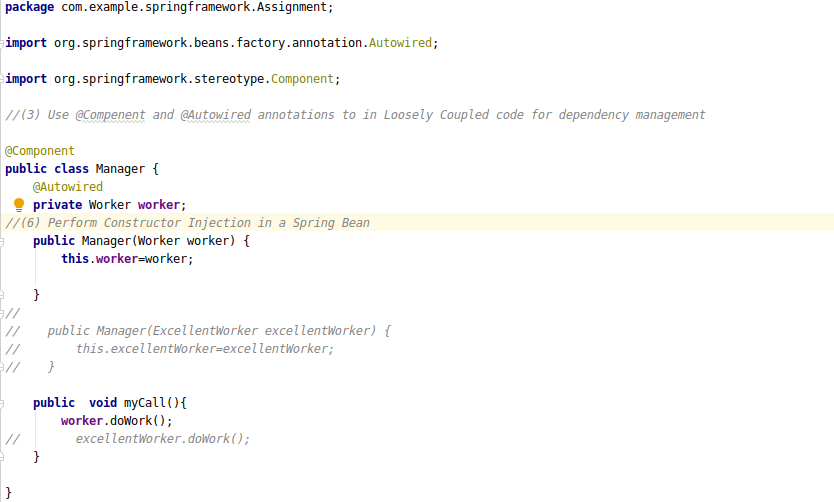




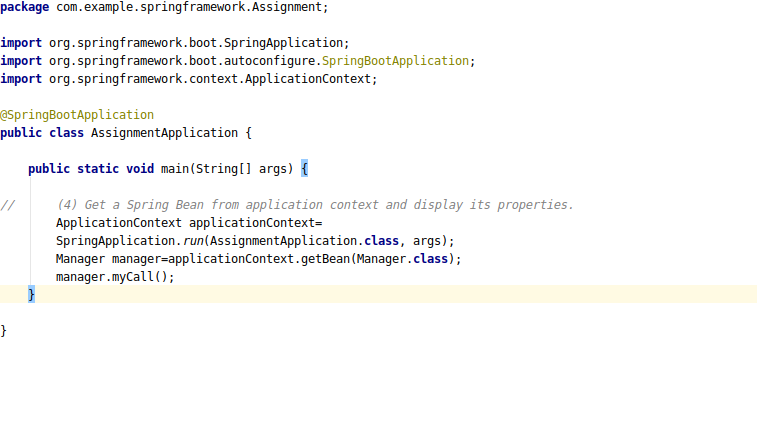
(3) Use @Compenent and @Autowired annotations to in Loosely Coupled code for dependency management

Ans:- The @Autowired annotation allows the container to automatically inject the *message* property from the *application-context.xml* file. The container searches for *String* bean object in the xml file and injects it here. If multiple String beans exist, then we will get an error.

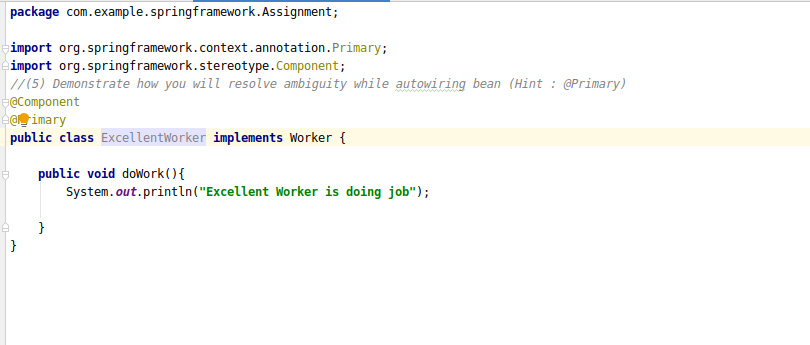
The @Component tells that *Message* is a component, a bean whose object has to be created. The *"msgObject"* is the name of the created object.

SS-

(4) Get a Spring Bean from application context and display its properties.

Ans:- 

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

SS-

(6) Perform Constructor Injection in a Spring Bean

SS-