**Important Instructions:**

* **At the end of assignment, you will able to implement DI, Bean Lifecycle and Spring bean scope.**
* **Your code will be graded both on correctness and efficiency.**
* **Use comments in your code that explains your assumptions and design decisions.**
* **You need to submit your assignment solution by end of the day.**
* **Before submitting your assignment makes sure it as per the given requirement.**
* **You need to submit your assignment on mercury on the respective folder.**
* **Your folder name will be your employeeId\_Day\_Assignment. (E.G 105678\_Day1\_Assignment).**

**Q1.**  **Make a Spring Application as per the below given specification:**

i) Design an Employee class having following members properties

eId int

name String

sal float

ii) Generate setter and getters for all properties

iii) **Declare a class MainApp having main method and bootstrap the Spring application using**

**BeanFactory and ApplicationContext both.**

**iv) Display the information using setter method.**

**Note : Use setter based dependency injection**

**Q2.**  **Make a Spring Application as per the below given specification:**

i) Design an Employee class having following members properties

eId int

name String

sal float

ii) Declare a parametrized constructor for initializing the properties

iii) Declare a method printInfo() to display all the information.

**iv) Declare a class MainApp having main method and bootstrap the Spring application using ApplicationContext**

**Note : Use constructor dependency injection**

**Q3.**  **Make a Spring Application as per the below given specification:**

i) Design an Employee class having following properties

eId int

name String

dept Department

ii) Design Department class having following properties

deptName String

deptLoc String

**iii) Declare a class MainApp having main method and bootstrap the Spring application using**

**ApplicationContext**

**iv) Display Employee and Department information.**

**Note : Use setter based as well as constructor based dependency injection**

**Q4.**  **Make a Spring Application as per the below given specification:**

i) Design an Employee class having following properties

eId int

name String

sal float

ii) Declare a default constructor

iii) Declare a parametrized constructor for initializing the properties

iv) Declare a static getData() method this method takes all the inputs form the keyboard and

return the Employee object.

v) Declare a method empInfo() which prints Employee details.

vi) Declare a class MainApp having main method and bootstrap the Spring application using

ApplicationContext

**Q5.**  **Make a Spring Application as per the below given specification:**

i) Design an Employee class having following properties

eId int

name String

sal float

ii) Declare a parametrized constructor for initializing the properties

iii) Generate setter and getter method.

iv) Initialize eId by providing value inside **spring.xml** file for rest of the properties use Spring Lifecycle method.

v) For using Lifecycle method use all three methods

1. Bean definition attribute inside XML file.
2. By Interface and
3. By Annotations

vi) Define bean scope as prototype at the time of bean definition.