# SPRING ASSIGNMENT Day_3

**Harshit Kushmakar| 16896**

## 1. Use @Autowire annotation to do the autowiring in Questions – 5 of Day 2 and explain with the help of code how autowiring is resolved in case of annotations.

```
package com.springcore;


import
org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class Main {
    public static void main(String[] args) {
        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext();
        context.scan("com.springcore");
        context.refresh();
        Customer cust = (Customer) context.getBean(Customer.class);
        cust.setCustomerName("Harshit");
        ClassAddress add = (ClassAddress)
context.getBean(ClassAddress.class);
        add.setCity("noida");
        System.out.println(cust);



    }
}
```

```
package com.springcore;

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

import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;
import java.time.LocalDate;
import java.util.List;
import java.util.Set;
@Component
public class Customer {
    private int customerId;
    private String customerName;
    private double monthlyIncome;
    private String profession;
    private String designation;
    private String companyName;
    private List<String> phoneNumbers;
    private Set<String> emailAdressSet;
    private LocalDate dateOfBirth;


   private List<LoanAgreement> loanAgreement;
     @Autowired
    private  ClassAddress classAddress;


    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public void setMonthlyIncome(double monthlyIncome) {
        this.monthlyIncome = monthlyIncome;
    }

    public void setProfession(String profession) {
        this.profession = profession;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }

    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    public void setPhoneNumbers(List<String> phoneNumbers) {
        this.phoneNumbers = phoneNumbers;
    }

    public void setEmailAdressSet(Set<String> emailAdressSet) {
        this.emailAdressSet = emailAdressSet;
    }

    public void setDateOfBirth(LocalDate dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }

    public void setLoanAgreement(List<LoanAgreement> loanAgreement) {
        this.loanAgreement = loanAgreement;
    }

    public void setClassAddress(ClassAddress classAddress) {
        this.classAddress = classAddress;
    }


    public Customer(int customerId, String customerName, double
monthlyIncome, String profession, String designation, String companyName,
List<String> phoneNumbers, Set<String> emailAdressSet, LocalDate
dateOfBirth, List<LoanAgreement> loanAgreement, ClassAddress classAddress)
{
        this.customerId = customerId;
        this.customerName = customerName;
        this.monthlyIncome = monthlyIncome;
        this.profession = profession;
        this.designation = designation;
        this.companyName = companyName;
        this.phoneNumbers = phoneNumbers;
        this.emailAdressSet = emailAdressSet;
        this.dateOfBirth = dateOfBirth;
        this.loanAgreement = loanAgreement;
        this.classAddress = classAddress;
    }

    @Override
    public String toString() {
        return "Customer{" +
                "customerId=" + customerId +
                ", customerName='" + customerName + '\'' +
                ", monthlyIncome=" + monthlyIncome +
                ", profession='" + profession + '\'' +
                ", designation='" + designation + '\'' +
                ", companyName='" + companyName + '\'' +
                ", phoneNumbers=" + phoneNumbers +
                ", emailAdressSet=" + emailAdressSet +
                ", dateOfBirth=" + dateOfBirth +
                ", loanAgreement=" + loanAgreement +
                ", classAddress=" + classAddress +
                '}';
    }

    public Customer() {

    }

    public void init(){
       for(int i =0; i<phoneNumbers.size(); i++) {
           String phoneno = phoneNumbers.get(i);
          if(!phoneno.startsWith("+91")){
              phoneNumbers.set(i,"+91"+ phoneno);
          }
        }
    }

    public void destroy(){
        phoneNumbers.clear();
        emailAdressSet.clear();
    }
}
```

```
package com.springcore;

import org.springframework.stereotype.Component;

@Component
public class ClassAddress {
    private int addressId;
    private String AddressLine1;
    private String AddressLine2;
    private String City;
    private String state;
    private int Zip;
```

```java
    public ClassAddress(int addressId, String addressLine1, String
addressLine2, String city, String state, int zip) {
        this.addressId = addressId;
        AddressLine1 = addressLine1;
        AddressLine2 = addressLine2;
        City = city;
        this.state = state;
        Zip = zip;
    }

    @Override
    public String toString() {
        return "ClassAddress{" +

                "addressId=" + addressId +

                ", AddressLine1='" + AddressLine1 + '\'' +

                ", AddressLine2='" + AddressLine2 + '\'' +

                ", City='" + City + '\'' +

                ", state='" + state + '\'' +

                ", Zip=" + Zip +
                '}';
    }

    public int getAddressId() {
        return addressId;
    }

    public void setAddressId(int addressId) {
        this.addressId = addressId;
    }

    public String getAddressLine1() {
        return AddressLine1;
    }

    public void setAddressLine1(String addressLine1) {
        AddressLine1 = addressLine1;
    }

    public String getAddressLine2() {
        return AddressLine2;
    }

    public void setAddressLine2(String addressLine2) {
        AddressLine2 = addressLine2;
    }

    public String getCity() {
        return City;
    }

    public void setCity(String city) {
        City = city;
    }

    public String getState() {
        return state;
    }

    public void setState(String state) {
        this.state = state;
    }

    public int getZip() {
        return Zip;
    }

    public void setZip(int zip) {
        Zip = zip;
    }

    public ClassAddress() {

    }
}
```

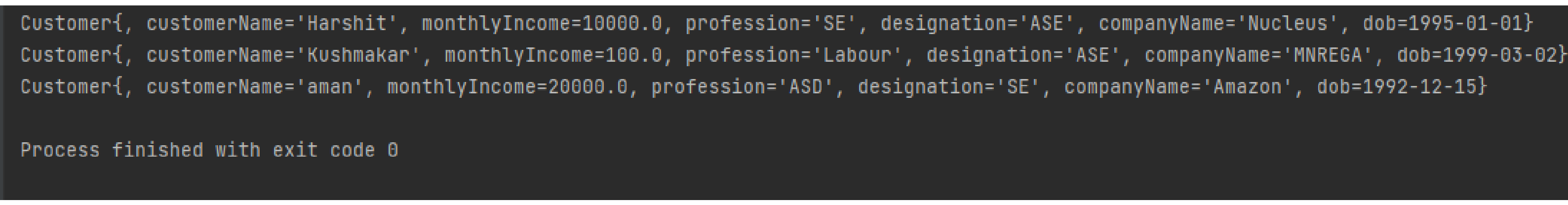
**2. Use Java Configuration to re-create the Questions 1, 2 of Spring Day-1 and Question – 2, 3 of Spring Day-2.**

```
package driver;

import Logger.MyLogger;
import appConfig.AppConfig;
import appConfig.AppConfiguration;

import model.Customer;
import org.springframework.context.ConfigurableApplicationContext;
import
org.springframework.context.annotation.AnnotationConfigApplicationContext;
import service.CustomerServiceImpl;

import java.sql.Connection;

public class MainConfiguration {
    public static void main(String[] args) {
        ConfigurableApplicationContext applicationContext=new
AnnotationConfigApplicationContext(AppConfiguration.class);
        ConfigurableApplicationContext context = new
AnnotationConfigApplicationContext(AppConfig.class);
        Customer user = (Customer) context.getBean("customer");
        Customer user2 = (Customer) context.getBean("customer2");
        Customer user3 = (Customer) context.getBean("customer3");
 System.out.println(user);
```

```java
System.out.println(user2);
System.out.println(user3);
```

```java
package appConfig;


import model.Address;
import model.Customer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Scope;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

@Configuration
@ComponentScan("model")

public class AppConfiguration {


    @Bean
    @Scope(value = "prototype")
    public Customer customerBean() {
        Customer customer = new Customer();
        customer.setCustomerName("HArshit");
        customer.setCustomerId(101);
        customer.setCompanyName("Nucleus");

        List<String> number = new ArrayList<>();
        number.add("8115552882");
        number.add("557463888");

        customer.setPhoneNumbers(number);

        Set<String> email = new HashSet<>();
        email.add("kush@gamil.com");
        email.add("shi@gamil.com");
        customer.setEmailAddressSet(email);
        DateTimeFormatter dateTimeFormatter =
DateTimeFormatter.ofPattern("dd/MM/yyyy");
        customer.setDob(LocalDate.parse("12/10/2000", dateTimeFormatter));

        return customer;
    }

    @Bean
    public Address address() {
        Address address = new Address();
        address.setAddressId(1);
        address.setAddressLine2("temp");
        address.setCity("city");

        return address;
    }

}
```

```java
package appConfig;


import model.Customer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;

import java.time.LocalDate;


@Configuration
@ComponentScan(basePackages="Model")
public class AppConfig {
    @Bean("customer")
    public Customer getCustomer() {
        return new Customer("Harshit", 10000, "SE", "ASE",
"Nucleus",LocalDate.of(1995, 01, 01));
    }

    @Bean("customer2")
    public Customer getCustomer2() {
        return new Customer("Kushmakar", 100, "Labour", "ASE", "MNREGA",
LocalDate.of(1999, 03, 02));
    }

    @Bean("customer3")
    public Customer getCustomer3() {
        return new Customer("aman", 20000, "ASD", "SE", "Amazon",
LocalDate.of(1992, 12, 15));

    }
}
```



```
Customer{, customerName='Harshit', monthlyIncome=10000.0, profession='SE', designation='ASE', companyName='Nucleus', dob=1995-01-01}
Customer{, customerName='Kushmakar', monthlyIncome=100.0, profession='Labour', designation='ASE', companyName='MNREGA', dob=1999-03-02}
Customer{, customerName='aman', monthlyIncome=20000.0, profession='ASD', designation='SE', companyName='Amazon', dob=1992-12-15}

Process finished with exit code 0
```

**3. Show the use of @ComponentScan annotation with the help of a demo code and explain about the different Annotations like @Component, @Service and @Repository using the comments.**

```java
package ComponentAnnotation;

// Importing required classes
import org.springframework.context.ApplicationContext;
import
org.springframework.context.annotation.AnnotationConfigApplicationContext;


public class Main {


    public static void main(String[] args)
    {


        ApplicationContext context
                = new AnnotationConfigApplicationContext(
                CollegeConfig.class);

        // Get the bean and storing it in
        // a object of College class type
        College college
                = context.getBean("collegeBean", College.class);

        // Invoking the method
        // inside main() method
        college.test();
    }
}
```

```java
package ComponentAnnotation;

import org.springframework.stereotype.Component;

@Component("collegeBean")

public class College {

    // Method
    public void test()
    {


        System.out.println("Test College Method");
    }
}
```

```java
package ComponentAnnotation;


import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;


@Configuration
```

```
@ComponentScan(basePackages = "ComponentAnnotation")


public class CollegeConfig {
}
```

## @Component

It is a class-level annotation that turns the class into Spring bean at the auto-scan time.
Example:
@Component
Public class Teachers
{
……
}
The component annotation can automatically detect custom beans. It represents that the framework could autodetect these classes for dependency injection.

## @Service

It is used at the class level. It shows that the annotated class is a service class, such as business basic logic, and call external APIs.
Example:
@Service
public class TestService
{
public void service1()
{
// business code
}
}
The @service annotation is used where the classes provide some business functionalities. The spring context autodetects these classes as the annotation is used with those classes where the business functionalities are to be used.

## Repository

It is a Data Access Object (DAO) that accesses the database directly. It indicates that the annotated class is a repository.
Example:

```
@Repository
public class TestRepository
{
public void delete()
{
// persistence code
}
}
```

The repository annotation indicates the class has the capability of storage, retrieval, updating, deletion, and search.