THEORY:

Spring Framework:

Spring is a lightweight framework. It can be thought of as a framework of frameworks because it provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF, etc. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems. The Spring framework comprises several modules such as IOC, AOP, DAO, Context, ORM, WEB MVC etc. We will learn these modules in next page. Let's understand the IOC and Dependency Injection first.

Dependency Injection in Spring:

Dependency Injection (DI) is a design pattern that removes the dependency from the programming code so that it can be easy to manage and test the application. Dependency Injection makes our programming code loosely coupled.

Autowiring in Spring:

Autowiring feature of spring framework enables you to inject the object dependency implicitly. It internally uses setter or constructor injection. Autowiring can't be used to inject primitive and string values. It works with reference only.

Advantage of Autowiring:

It requires the **less code** because we don't need to write the code to inject the dependency explicitly.

Disadvantage of Autowiring:

No control of programmer. It can't be used for primitive and string values.

Autowiring Modes:

There are many autowiring modes:

No.	Mode	Description
1)	no	It is the default autowiring mode. It means no autowiring bydefault.
2)	byName	The byName mode injects the object dependency according to name of the bean. In such case, property name and bean name must be same. It internally calls setter method.
3)	byType	The byType mode injects the object dependency according to type. So property name and bean name can be different. It internally calls setter method.

4)	constructor	The constructor mode injects the dependency by calling the constructor of the class. It calls the constructor having large number of parameters.
5)	autodetect	It is deprecated since Spring 3.

A) Write a program to print “Hello World” using spring framework.

SOURCE CODE:

Message.java:

```
public class Message {

public void printMessage() {
            System.out.println("Hello World Spring-JAVA from Narender Keswani");
            }
}
```

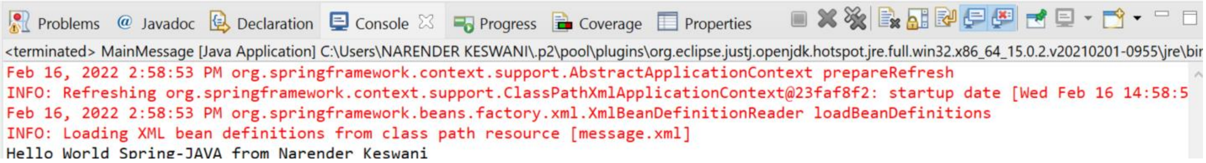
message.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="messageBean" class="Message">
</bean>
</beans>
```

MainMessage.java:

```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainMessage {

public static void main(String[] args) {
            // TODO Auto-generated method stub
            ApplicationContext context1= new ClassPathXmlApplicationContext("message.xml");
            Message msg= (Message) context1.getBean("messageBean");
            msg.printMessage();
}
}
```

OUTPUT:



```
<terminated> MainMessage [Java Application] C:\Users\NARENDER KESWANI\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bir
Feb 16, 2022 2:58:53 PM org.springframework.context.support.AbstractApplicationContext prepareRefresh
INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@23faf8f2: startup date [Wed Feb 16 14:58:5
Feb 16, 2022 2:58:53 PM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
INFO: Loading XML bean definitions from class path resource [message.xml]
Hello World Spring-JAVA from Narender Keswani
```

B) Write a program create Car class and Engine class circle class Has-A relationship with Engine class. Use constructor injection to invoke object of Engine class in car class

SOURCE CODE:

Engine.java:

```
public class Engine {
private int engineNo;
private String engineName;
private String engineType;


public Engine(int engineNo, String engineName, String engineType) {
		super();
		this.engineNo = engineNo;
		this.engineName = engineName;
		this.engineType = engineType;
}


public String engineDetails() {
		return "\nEngine Details: \nEngine Number = "+ engineNo + "\nEngine Name = "+
engineName + "\nEngine Type = " + engineType;
		}
}
```

Car.java:

```
public class Car {

private String name;
private int price;
private int seats;
private String color;
private String fuelType;
private Engine engine;


public Car(String name, int price, int seats, String color, String fuelType, Engine engine) {
		super();
		this.name = name;
		this.price = price;
		this.seats = seats;
		this.color = color;
		this.fuelType = fuelType;
		this.engine = engine;
}


public void carDetails() {
```

```
            System.out.println("\nCar Details: \nCar Name = "+ name + "\nCar Price = "+ price +
"\nCar Seats = " + seats + "\nCar Color = "+ color + "\nCar Fuel Type = " + fuelType);
            System.out.println(engine.engineDetails());
            }
}
```

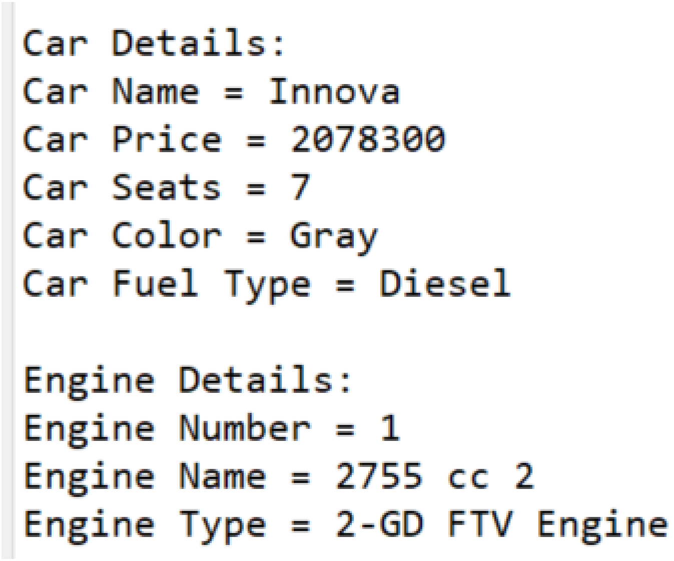
CarEngineCDI.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id = "engineBean" class = "Engine">
<constructor-arg value = "1" type = "int"></constructor-arg>
<constructor-arg value="2755 cc 2"></constructor-arg>
<constructor-arg value="2-GD FTV Engine"></constructor-arg>
</bean>
<bean id = "carBean" class = "Car">
<constructor-arg value="Innova"></constructor-arg>
<constructor-arg value="2078300" type="int"></constructor-arg>
<constructor-arg value="7" type="int"></constructor-arg>
<constructor-arg value="Gray"></constructor-arg>
<constructor-arg value="Diesel"></constructor-arg>
<constructor-arg><ref bean="engineBean"/></constructor-arg>
</bean>
</beans>
```

MainCarConstructor.java:

```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainCarConstructorDI {

public static void main(String[] args) {
            // TODO Auto-generated method stub
ApplicationContext context= new ClassPathXmlApplicationContext("CarEngineCDI.xml");
Car c= (Car) context.getBean("carBean");
c.carDetails();
}
}
```

OUTPUT:



```
Car Details:
Car Name = Innova
Car Price = 2078300
Car Seats = 7
Car Color = Gray
Car Fuel Type = Diesel

Engine Details:
Engine Number = 1
Engine Name = 2755 cc 2
Engine Type = 2-GD FTV Engine
```

C) Write a program create Car class and Engine class circle class Has-A relationship with Engine class. Use setter injection to invoke object of Engine class in car class

SOURCE CODE:

Engine.java:

```
public class Engine {
private int engineNo;
private String engineName;
private String engineType;

public Engine() {
}

public Engine(int engineNo, String engineName, String engineType) {
        super();
        this.engineNo = engineNo;
        this.engineName = engineName;
        this.engineType = engineType;
}

public int getEngineNo() {
        return engineNo;
}

public void setEngineNo(int engineNo) {
        this.engineNo = engineNo;
}

public String getEngineName() {
        return engineName;
}

public void setEngineName(String engineName) {
        this.engineName = engineName;
}

public String getEngineType() {
        return engineType;
}

public void setEngineType(String engineType) {
        this.engineType = engineType;
}

public String engineDetails() {
        return "\nEngine Details: \nEngine Number = " + engineNo + "\nEngine Name = " +
engineName + "\nEngine Type = "
                        + engineType;
}
}
```

Car.java:

```
public class Car {

private String name;
private int price;
private int seats;
private String color;
private String fuelType;
private Engine engine;

public Car() { }


public Car(String name, int price, int seats, String color, String fuelType, Engine engine) {
	super();
	this.name = name;
	this.price = price;
	this.seats = seats;
	this.color = color;
	this.fuelType = fuelType;
	this.engine = engine;
}

public String getName() {
	return name;
}


public void setName(String name) {
	this.name = name;
}


public int getPrice() {
	return price;
}


public void setPrice(int price) {
	this.price = price;
}


public int getSeats() {
	return seats;
}
public void setSeats(int seats) {
		this.seats = seats;
}


public String getColor() {
		return color;
}


public void setColor(String color) {
		this.color = color;
}


public String getFuelType() {
		return fuelType;
}


public void setFuelType(String fuelType) {
		this.fuelType = fuelType;
}


public Engine getEngine() {
		return engine;
}


public void setEngine(Engine engine) {
		this.engine = engine;
}


public void carDetails() {
		System.out.println("\nCar Details: \nCar Name = "+ name + "\nCar Price = "+ price +
"\nCar Seats = " + seats + "\nCar Color = "+ color + "\nCar Fuel Type = " + fuelType);
		System.out.println(engine.engineDetails());
		}
}
```

CarEngineSDI.xml:

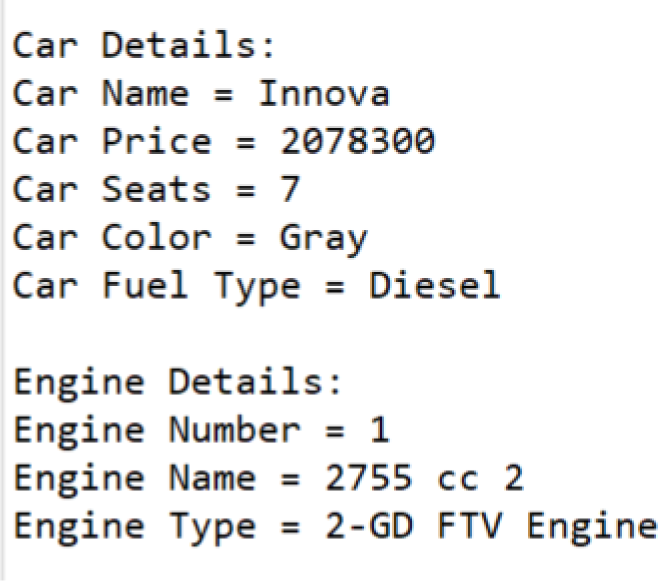
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id = "engineBean" class = "Engine">
<property name="engineNo" value = "1"></property>
<property name="engineName" value="2755 cc 2"></property>
```

```
<property name="engineType" value="2-GD FTV Engine"></property>
</bean>
<bean id = "carBean" class = "Car">
<property name="name" value="Innova"></property>
<property name="price" value="2078300"></property>
<property name="seats" value="7"></property>
<property name="color" value="Gray"></property>
<property name="fuelType" value="Diesel"></property>
<property name="Engine" ref="engineBean"></property>
</bean>
</beans>
```

MainCarSetterDI.java:

```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainCarSetterDI {

public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext context= new
ClassPathXmlApplicationContext("CarEngineSDI.xml");
        Car c= (Car) context.getBean("carBean");
        c.carDetails();
}
}
```

OUTPUT:



```
Car Details:
Car Name = Innova
Car Price = 2078300
Car Seats = 7
Car Color = Gray
Car Fuel Type = Diesel

Engine Details:
Engine Number = 1
Engine Name = 2755 cc 2
Engine Type = 2-GD FTV Engine
```

D) Write a program create Car class and Engine class circle class Has-A relationship with Engine class. Use Auto-wiring through XML (any one byname/bytype/by constructor) for dependency injection

SOURCE CODE:

Engine:

```
package xmlbytype;

public class Engine {
private int engineNo;
private String engineName;
private String engineType;

public Engine() {
}

public Engine(int engineNo, String engineName, String engineType) {
        super();
        this.engineNo = engineNo;
        this.engineName = engineName;
        this.engineType = engineType;
}

public int getEngineNo() {
        return engineNo;
}

public void setEngineNo(int engineNo) {
        this.engineNo = engineNo;
}

public String getEngineName() {
        return engineName;
}

public void setEngineName(String engineName) {
        this.engineName = engineName;
}

public String getEngineType() {
        return engineType;
}

public void setEngineType(String engineType) {
        this.engineType = engineType;
}

public String engineDetails() {
```

```
            return "\nEngine Details: \nEngine Number = " + engineNo + "\nEngine Name = " +
engineName + "\nEngine Type = "
                                    + engineType;
}

}
```

Car.java:

```
package xmlbytype;

public class Car {
private Engine engine;

public Car() { }

public Engine getEngine() {
            return engine;
}

public void setEngine(Engine engine) {
            this.engine = engine;
}

public void carDetails() {
            System.out.println(engine.engineDetails());
            }
}
```

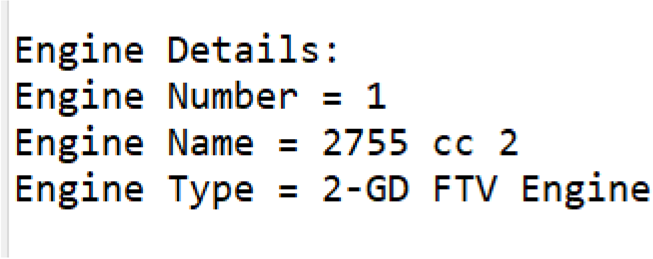
CarEngineByTypeXML.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:context="http://www.springframework.org/schema/context"
     xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.1.xsd">

<bean id = "car" class = "xmlbytype.Car" autowire="byType"/>

<bean id="engine" class = "xmlbytype.Engine">
<property name="engineNo" value = "1"></property>
<property name="engineName" value="2755 cc 2"></property>
<property name="engineType" value="2-GD FTV Engine"></property>
</bean>
</beans>
```

MainCarEngineByTypeXML:

```
package xmlbytype;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;


public class MainCarEngineByTypeXML {

public static void main(String[] args) {
		// TODO Auto-generated method stub
		ApplicationContext context= new 
ClassPathXmlApplicationContext("CarEngineByTypeXML.xml");
		Car c= (Car) context.getBean("car");
		c.carDetails();
}

}
```

OUTPUT:



```
Engine Details:
Engine Number = 1
Engine Name = 2755 cc 2
Engine Type = 2-GD FTV Engine
```

E) Write a program create Car class and Engine class circle class Has-A relationship with Engine class. Use @autowiring to invoke dependency injection (any two method)

1) AutoWired (byName)

SOURCE CODE:

Car.java:

```
package autowiring;
import org.springframework.beans.factory.annotation.Autowired;


public class Car {

    private String name;
    private int price;
    private int seats;
    private String color;
    private String fuelType;

    //Note this Autowiring [By Name of object it will be mapped]
    @Autowired
    private Engine engine;

    public Car() { }

    public Car(String name, int price, int seats, String color, String fuelType, Engine
engine) {
            super();
            this.name = name;
            this.price = price;
            this.seats = seats;
            this.color = color;
            this.fuelType = fuelType;
            this.engine = engine;
    }

    public String getName() {
            return name;
    }


    public void setName(String name) {
            this.name = name;
    }


    public int getPrice() {
            return price;
    }
```

```
	public void setPrice(int price) {
		this.price = price;
	}


	public int getSeats() {
		return seats;
	}


	public void setSeats(int seats) {
		this.seats = seats;
	}


	public String getColor() {
		return color;
	}


	public void setColor(String color) {
		this.color = color;
	}


	public String getFuelType() {
		return fuelType;
	}


	public void setFuelType(String fuelType) {
		this.fuelType = fuelType;
	}


	public Engine getEngine() {
		return engine;
	}


	public void setEngine(Engine engine) {
		this.engine = engine;
	}


	public void carDetails() {
		System.out.println("\nCar Details: \nCar Name = "+ name + "\nCar Price = "+
price + "\nCar Seats = " + seats + "\nCar Color = "+ color + "\nCar Fuel Type = " +
fuelType);
```

```
                System.out.println(engine.engineDetails());
                }
}
```

Engine.java:

```
package autowiring;

public class Engine {
    private int engineNo;
    private String engineName;
    private String engineType;

    public Engine() {
    }

    public Engine(int engineNo, String engineName, String engineType) {
            super();
            this.engineNo = engineNo;
            this.engineName = engineName;
            this.engineType = engineType;
    }

    public int getEngineNo() {
            return engineNo;
    }

    public void setEngineNo(int engineNo) {
            this.engineNo = engineNo;
    }

    public String getEngineName() {
            return engineName;
    }

    public void setEngineName(String engineName) {
            this.engineName = engineName;
    }

    public String getEngineType() {
            return engineType;
    }

    public void setEngineType(String engineType) {
            this.engineType = engineType;
    }

    public String engineDetails() {
            return "\nEngine Details: \nEngine Number = " + engineNo + "\nEngine
Name = " + engineName + "\nEngine Type = "
                            + engineType;
```

```
        }
    }
```

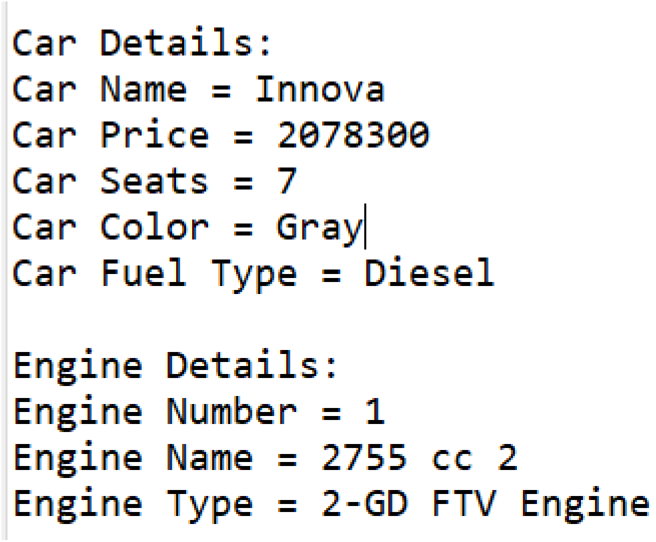
CarEngineByNameAutoWired.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="
      http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
      http://www.springframework.org/schema/context
      http://www.springframework.org/schema/context/spring-context-3.1.xsd">
<context:annotation-config/>
<bean name="engine" class = "autowiring.Engine">
<property name="engineNo" value = "1"></property>
<property name="engineName" value="2755 cc 2"></property>
<property name="engineType" value="2-GD FTV Engine"></property>
</bean>
<bean name = "car" class = "autowiring.Car">
<property name="name" value="Innova"></property>
<property name="price" value="2078300"></property>
<property name="seats" value="7"></property>
<property name="color" value="Gray"></property>
<property name="fuelType" value="Diesel"></property>
</bean>
</beans>
```

MainCarEngineByNameAutowiring.java:

```
package autowiring;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainCarEngineByNameAutowiring {

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                ApplicationContext context= new
ClassPathXmlApplicationContext("CarEngineByNameAutowired.xml");
                Car c= (Car) context.getBean("car");
                c.carDetails();
        }

}
```

OUTPUT:



```
Car Details:
Car Name = Innova
Car Price = 2078300
Car Seats = 7
Car Color = Gray
Car Fuel Type = Diesel

Engine Details:
Engine Number = 1
Engine Name = 2755 cc 2
Engine Type = 2-GD FTV Engine
```

2) AutoWired(byType)

SOURCE CODE:

Engine:

```
package autowiredbytype;

public class Engine {
    private int engineNo;
    private String engineName;
    private String engineType;

    public Engine() {
    }

    public Engine(int engineNo, String engineName, String engineType) {
            super();
            this.engineNo = engineNo;
            this.engineName = engineName;
            this.engineType = engineType;
    }

    public int getEngineNo() {
            return engineNo;
    }

    public void setEngineNo(int engineNo) {
            this.engineNo = engineNo;
    }

    public String getEngineName() {
            return engineName;
    }

    public void setEngineName(String engineName) {
            this.engineName = engineName;
    }

    public String getEngineType() {
            return engineType;
    }

    public void setEngineType(String engineType) {
            this.engineType = engineType;
    }

    public String engineDetails() {
            return "\nEngine Details: \nEngine Number = " + engineNo + "\nEngine
Name = " + engineName + "\nEngine Type = "
                        + engineType;
    }

}
```

Car.java:

```
package autowiredbytype;

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

public class Car {

    @Autowired
    private Engine engine;

    public Car() { }

    public Engine getEngine() {
            return engine;
    }

    public void setEngine(Engine engine) {
            this.engine = engine;
    }

    public void carDetails() {
            System.out.println(engine.engineDetails());
            }

}
```

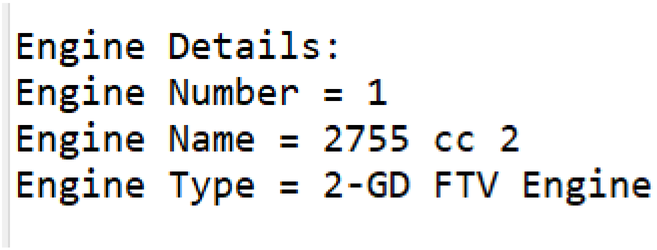
CarEngineByTypeAutoWired.xml:

```
  <?xml version="1.0" encoding="UTF-8"?>
  <beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:context="http://www.springframework.org/schema/context"
      xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.1.xsd">
  <context:annotation-config/>
  <bean name="engine" class = "autowiredbytype.Engine">
  <property name="engineNo" value = "1"></property>
  <property name="engineName" value="2755 cc 2"></property>
  <property name="engineType" value="2-GD FTV Engine"></property>
  </bean>
  <bean name = "car" class = "autowiredbytype.Car"/>
  </beans>
```

MainCarEngineAutoWiredByType:

```
package autowiredbytype;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainCarEngineAutoWiredByType {

   public static void main(String[] args) {
              // TODO Auto-generated method stub
                              ApplicationContext context= new
ClassPathXmlApplicationContext("CarEngineByTypeAutowired.xml");
                              Car c= (Car) context.getBean("car");
                              c.carDetails();
   }

}
```

OUTPUT:



```
Engine Details:
Engine Number = 1
Engine Name = 2755 cc 2
Engine Type = 2-GD FTV Engine
```

CONCLUSION:

From this practical, I have learned about spring , dependency injection using setter and constructor, autowiring in XML & @Autowired annotation.