



Practical No:	12
Roll No:	1525
Title of Program:	Circular Dependency
Objective	Resolve circular dependency using setter injection and lazy-init.

CODE:

ClassA.java

```
package edu.met.p1;
```

```
public class ClassA {  
    ClassB b;  
    public ClassA(ClassB b)  
    {  
        this.b=b;  
    }  
}
```

ClassB.java

```
package edu.met.p1;
```

```
public class ClassB {  
    ClassA a;  
    public ClassB(ClassA a)  
    {  
        this.a=a;  
    }  
}
```

```
}
```

ClassC.java

```
package edu.met.p1;
```

```
public class ClassC {  
    ClassD d;  
    public ClassC()  
    {  
        System.out.println("Constructor C");  
    }  
    public void setD(ClassD d)  
    {  
        System.out.println("Inside Class C");  
        this.d=d;  
    }  
}
```

ClassD.java

```
package edu.met.p1;
```

```
public class ClassD {  
    ClassC c;  
    public ClassD() {  
        // TODO Auto-generated constructor stub  
        System.out.println("Constructor D");  
    }  
    public void setC(ClassC c)  
    {  
        System.out.println("Inside Class C");  
        this.c=c;  
    }  
}
```

```
}  
  
}
```

Appctx.xml

```
<beans  
  xmlns="http://www.springframework.org/schema/beans"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:p="http://www.springframework.org/schema/p"  
  xsi:schemaLocation="http://www.springframework.org/schema/beans  
    http://www.springframework.org/schema/beans/spring-beans-  
3.0.xsd">  
  <bean id="beanA" class="edu.met.p1.ClassA" lazy-init="true">  
    <constructor-arg name="b" ref="beanB"></constructor-arg>  
  </bean>  
  <bean id="beanB" class="edu.met.p1.ClassB" lazy-init="true">  
    <constructor-arg name="a" ref="beanA"></constructor-arg>  
  </bean>  
  <bean id="beanC" class="edu.met.p1.ClassC">  
    <property name="d" ref="beanD"></property>  
  </bean>  
  <bean id="beanD" class="edu.met.p1.ClassD">  
    <property name="c" ref="beanC"></property>  
  </bean>  
</beans>
```

CircularTest.java

```
package edu.met.p1;  
import org.springframework.context.ApplicationContext;  
import  
org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class CircularTest {  
    static ApplicationContext ctx;
```

```

public static void main(String[] args) {
    // TODO Auto-generated method stub
    ctx=new ClassPathXmlApplicationContext("Appctx.xml");
    //ClassA a1 = (ClassA)ctx.getBean("beanA");
    //ClassB b1 = (ClassB)ctx.getBean("beanB");
    ClassC c1=(ClassC)ctx.getBean("beanC");
    ClassD d1=(ClassD)ctx.getBean("beanD");
}
}

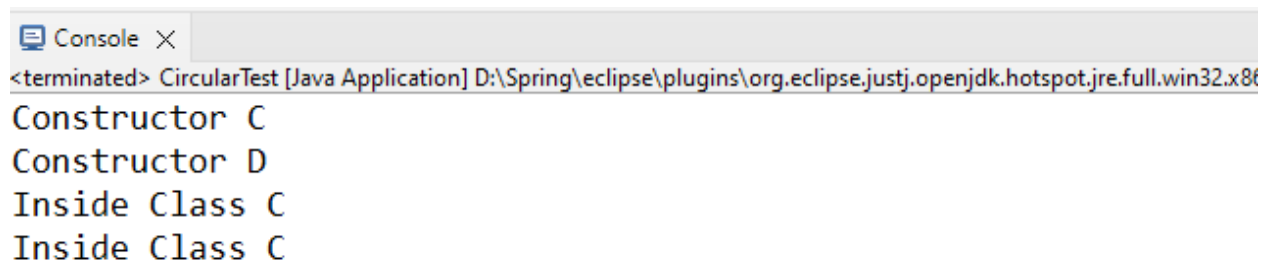
```

OUTPUT:

Error:

Error creating bean with name 'beanA': Requested bean is currently in creation: Is there an unresolvable circular reference?

Resolving circular dependency using setter injection and lazy-init



```

<terminated> CircularTest [Java Application] D:\Spring\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jdk\bin\java.exe
Constructor C
Constructor D
Inside Class C
Inside Class C

```