1. Write a JSP that takes the user’s name and age from a

form. -Echo backs the name and age along with a message

stating the price of movie tickets. -The price is determined

by the age passed to the JSP. -If the age is greater than 62,

the movie ticket price is $7.00. -If the user is less than 10

years old, the price is $5.00. -For everyone else, the price is $9.50.

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>page 1</title>

</head>

<body>

<form action=*"problem1Page.jsp"*>

Enter your name:<input type=*"text"* name=*"username"*><br>

Enter your age:<input type=*"number"* name=*"age"*><br>

<input type=*"submit"* value=*"submit"*>

</form>

</body>

</html>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<% **int** age=Integer.parseInt(request.getParameter("age"));

String name=request.getParameter("username");

**double** price;

**if**(age>62)

price=7.00;

**else** **if**(age<10)

price=5.00;

**else**

price=9.50;

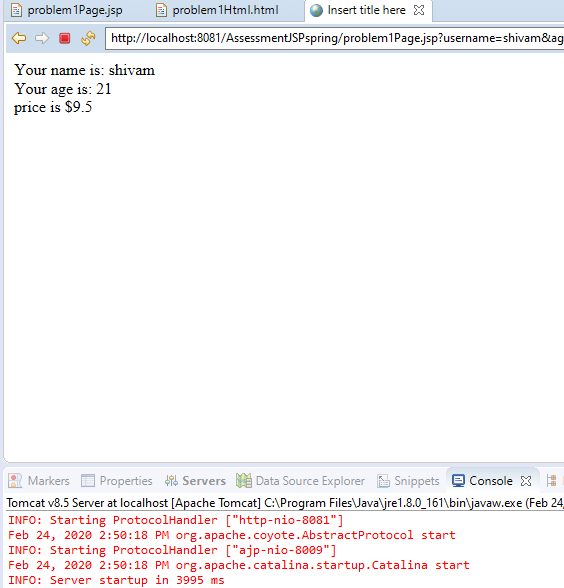
out.println("Your name is: "+name);%><br>

<%out.println("Your age is: "+age);%><br>

<%out.println("price is $"+price);%>

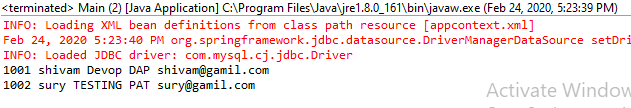
</body>

</html>



5. Write a spring jdbc program to display all the records from any table from the mysql database.

import java.util.Iterator;  
import java.util.List;  
   
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
   
   
   
public class Testprog5  
{  
   
     public static void main(String[] args)  
     {  
           
             
              ApplicationContext ctx=new ClassPathXmlApplicationContext("application-context.xml");  
           
              EmployeeDAOImpl dao=(EmployeeDAOImpl)ctx.getBean("edaoimpl");  
               
              List<Employee> employee =dao.getEmployee();  
                 
              Iterator<Employee> itr=employee.iterator();  
              for(Employee e:employee)  
              {  
                   System.out.println(e.getName()+" " +e.getId()+" "+e.getDept()+" "+e.getDesg()+" "+e.getCompany()+" "+e.getEmail());  
              }  
           
              dao.getEmployee();  
   
               
     }  
   
}  
   
Employee.java  
package springSBA;  
   
public class Employee  
{  
     private String name;  
     private String id;  
     private String dept;  
     private String desg;  
     private String company;  
     private String email;  
     
     
     public Employee() {  
          super();  
     }  
   
     public Employee(String name, String id, String dept, String desg, String company, String email) {  
          super();  
          [this.name](http://this.name/) = name;  
          [this.id](http://this.id/) = id;  
          this.dept = dept;  
          this.desg = desg;  
          this.company = company;  
          this.email = email;  
     }  
     
     public String getName() {  
          return name;  
     }  
     public void setName(String name) {  
          [this.name](http://this.name/) = name;  
     }  
     public String getId() {  
          return id;  
     }  
     public void setId(String id) {  
          [this.id](http://this.id/) = id;  
     }  
     public String getDept() {  
          return dept;  
     }  
     public void setDept(String dept) {  
          this.dept = dept;  
     }  
     public String getDesg() {  
          return desg;  
     }  
     public void setDesg(String desg) {  
          this.desg = desg;  
     }  
     public String getCompany() {  
          return company;  
     }  
     public void setCompany(String company) {  
          this.company = company;  
     }  
     public String getEmail() {  
          return email;  
     }  
     public void setEmail(String email) {  
          this.email = email;  
     }  
}  
   
employeedaoimpl  
   
package springSBA;  
   
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.List;  
   
import org.springframework.jdbc.core.JdbcTemplate;  
import org.springframework.jdbc.core.RowMapper;  
   
   
public class EmployeeDAOImpl implements EmployeeDAO  
{  
     private JdbcTemplate jdbcTemplate;  
       
     public void setJdbcTemplate(JdbcTemplate jdbcTemplate)  
     {  
         this.jdbcTemplate = jdbcTemplate;  
     }  
     
     
   
     @Override  
     public List<Employee> getEmployee() {  
          String sql="select \* from employee";  
          List<Employee> list=jdbcTemplate.query(sql, new RowMapper<Employee>(){  
   
              public Employee mapRow(ResultSet rs, int rowNum) throws SQLException  
              {  
           
                   Employee e=new Employee();  
                   e.setName(rs.getString("name"));  
                   e.setId(rs.getString("id"));  
                   e.setDept(rs.getString("dept"));  
                   e.setDesg(rs.getString("desg"));  
                   e.setCompany(rs.getString("company"));  
                   e.setEmail(rs.getString("email"));  
                   return e;  
                   
              }  
          });  
          return list;  
     }  
     }  
   
   
EmployeeDAO.java  
   
package springSBA;  
   
import java.util.List;  
   
public interface EmployeeDAO  
{  
     public List<Employee> getEmployee();  
}  
   
application-context.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: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="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">  
   
<property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />  
<property name="url" value="jdbc:mysql://localhost:3306/jdbc" />  
<property name="username" value="root" />  
<property name="password" value="root" />  
</bean>  
   
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">  
<property name="dataSource" ref="ds"></property>  
</bean>  
   
<bean id="edaoimpl" class="springSBA.EmployeeDAOImpl">  
<property name="jdbcTemplate" ref="jdbcTemplate"></property>  
</bean>  
   
</beans>



4. Write a simple spring program to implement Dependency injection using constructor method for dependent objects and Map objects.

Program:

Construct.java

package springexamples;

public class construct {

public construct(String value,int i)

{

System.out.println("hei i m from parameterized constructor.....");

}

}

Constructbean.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="hello" class="springexamples.construct">

<constructor-arg value="10"/>

<constructor-arg value="10" />

</bean>

</beans>

Main.java:

package jdbc;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class main {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext context=new ClassPathXmlApplicationContext("construcbean.xml");

context.getBean("hello");

}

}

3. Write a spring program which will demonstrates the spring life cycle bean post processor methods.

Program:

HelloWorld.java:

package BeanPost;

public class HelloWorld {

private String message;

public void setMessage(String message){

this.message=message;

}

public void getMessage(){

System.out.println("Your Message : "+ message);

}

public void init(){

System.out.println("Bean is going through init");

}

public void destroy(){

System.out.println("Bean will destroy now");

}

}

InitHelloWorld.java:

package BeanPost;

import org.springframework.beans.factory.config.BeanPostProcessor;

import org.springframework.beans.BeansException;

public class InitHelloWorld implements BeanPostProcessor {

public Object postProcessBeforeInitialization(Object bean, String beanName)

throws BeansException {

System.out.println("Before Initialization : " + beanName);

return bean;

}

public Object postProcessAfterInitialization(Object bean, String beanName)

throws BeansException {

System.out.println("After Initialization : " + beanName);

return bean;

}

}

Bean.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-3.0.xsd">

<bean id="helloWorld" class="BeanPost.HelloWorld"

init-method="init" destroy-method="destroy">

<property name="message" value="Hello World!"/>

</bean>

<bean class="BeanPost.InitHelloWorld" />

</beans>

Main.java :

package BeanPost;

import org.springframework.context.support.AbstractApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

public static void main(String[] args) {

AbstractApplicationContext context = new ClassPathXmlApplicationContext("Bean.xml");

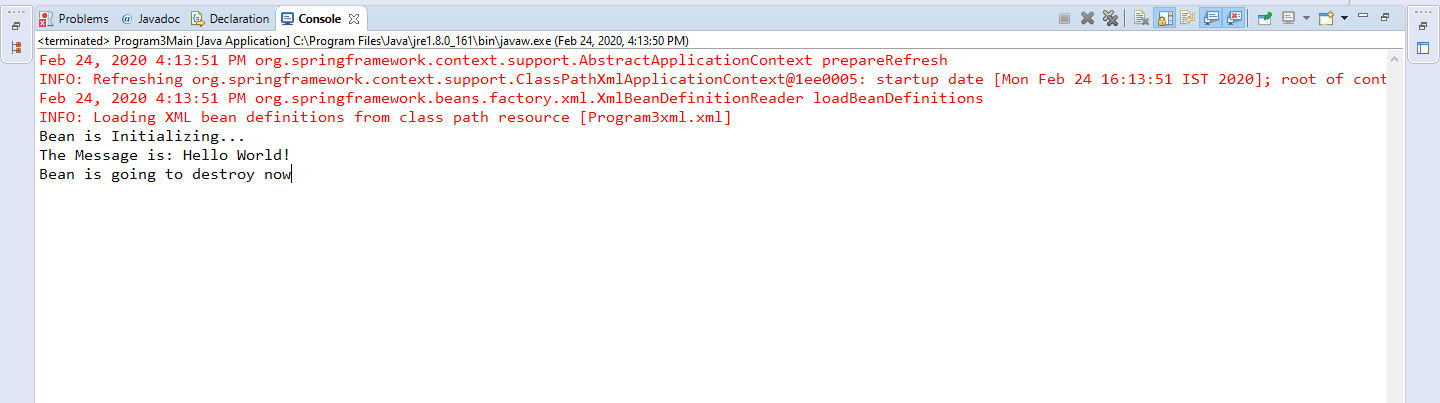
HelloWorld a = (HelloWorld) context.getBean("helloWorld");

a.getMessage();

context.registerShutdownHook();

}

}



2. Write a program using jsp and JDBC for developing an online

application for the shopping of computer science books. (Hint: use

concept of session tracking) You have to create a database for book

title, author(s) of book, publisher, year of publication, price.

Make necessary assumptions for book shopping.

Your application has to perform the following activities:

a. Display all books available

b. Select the books and add to cart

c. Once checkout is done, display the selected books along with total.

Sql:

create table book(book\_title varchar(20),author\_of\_book varchar(20),publisher varchar(20),year\_of\_publication bigint(10),price bigint(10));

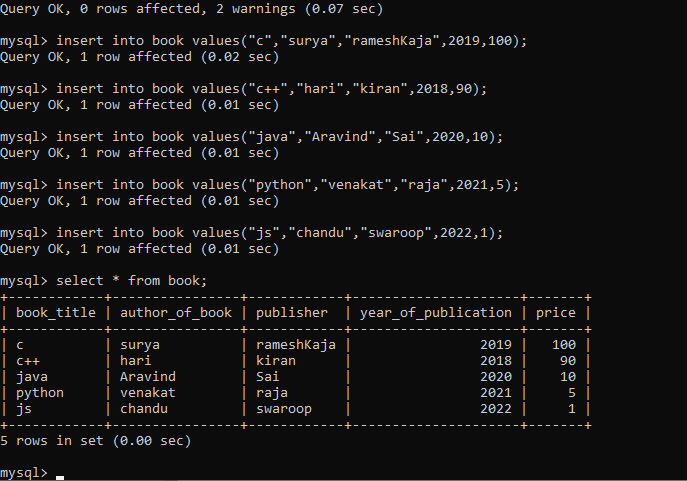
insert into book values("c","surya","rameshKaja",2019,100);

insert into book values("c++","hari","kiran",2018,90);

insert into book values("java","Aravind","Sai",2020,10);

insert into book values("python","venakat","raja",2021,5);

insert into book values("js","chandu","swaroop",2022,1);



<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>book page</title>

</head>

<body>

<a href=*"problem2display.jsp"*>Display</a><br>

<a href=*"problem2add.jsp"*>AddtoCart</a><br>

<a href=*"problem2checkout.jsp"*>Checkout</a><br>

</body>

</html>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"* import=*"java.util.\*"* import=*"java.sql.\*"*

import=*"javax.servlet.\*"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<%

DriverManager.registerDriver(**new** com.mysql.cj.jdbc.Driver());

Connection connection=DriverManager.

getConnection("jdbc:mysql://localhost:3306/cts","root","root");

String querry="select \* from book";

PreparedStatement pst=connection.prepareStatement(querry);

ResultSet rs=pst.executeQuery();

**while**(rs.next())

{

%>

<h3>Books are....</h3><br>

<form action=*"problem2add.jsp"*>

Book\_title<input type=*"text"* name=*"book\_title"*

value=*"*<%=rs.getString(1)%>*"*><br>

Author\_of\_book<input type=*"text"* name=*"author\_of\_book"*

value=*"*<%=rs.getString(2)%>*"*><br>

Publisher<input type=*"text"* name=*"publisher"*

value=*"*<%=rs.getString(3)%>*"*><br>

Year\_of\_publication<input type=*"text"* name=*"year\_of\_publication"*

value=*"*<%=rs.getInt(4)%>*"*><br>

Price<input type=*"text"* name=*"price"*

value=*"*<%=rs.getInt(5)%>*"*><br>

<input type=*"submit"* value=*"add"*>

</form>

<%}

pst.close();

connection.close();

%>

</body>

</html>

