AJAVA PRACTICAL

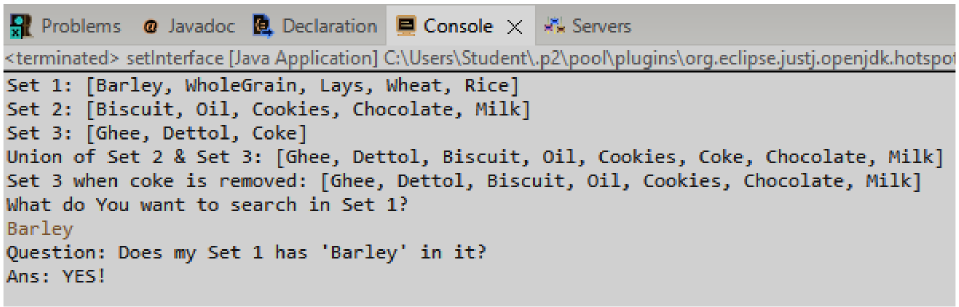
Q1 A) Set Interface containing list of items grocery and perform foll operation:
a.Add items in set
b.Insert items of one set into other set
c.Remove items from set
d.Search specified item in set

Code:

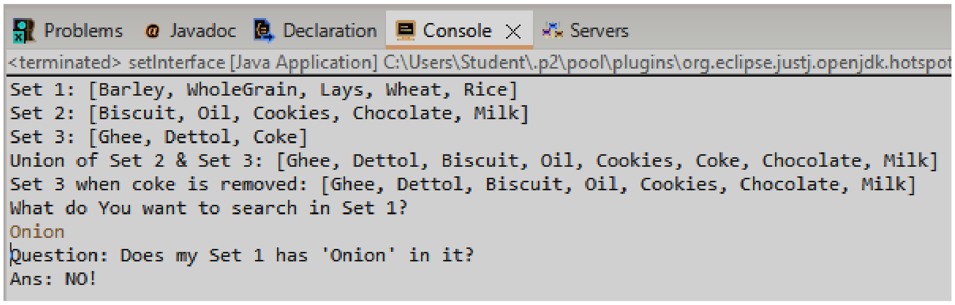
```
package sakshi;
import java.util.*;
public class setInterface {
        public static void main(String args[]) {
                Scanner sc = new Scanner(System.in);
                Set<String> set1 = new HashSet<>();
                set1.add("Wheat");
                set1.add("Rice");
                set1.add("Lays");
                set1.add("WholeGrain");
                set1.add("Barley");
                System.out.println("Set 1: " + set1);
                Set<String> set2 = new HashSet<>();
                set2.add("Milk");
                set2.add("Chocolate");
                set2.add("Cookies");
                set2.add("Biscuit");
                set2.add("Oil");
                System.out.println("Set 2: " + set2);
                Set<String> set3 = new HashSet<>();
                set3.add("Dettol");
                set3.add("Ghee");
                set3.add("Coke");
                System.out.println("Set 3: " + set3);
                set2.addAll(set3);
                System.out.println("Union of Set 2 & Set 3: " + set2);
                set2.remove("Coke");
                System.out.println("Set 3 when coke is removed: " + set2);
                System.out.println("What do You want to search in Set 1?");
                String searchText = sc.next();
                System.out.println("Question: Does my Set 1 has '" + searchText + "' in it?");
                if (set1.contains(searchText)) {
                        System.out.println("Ans: YES!");
                } else {
                        System.out.println("Ans: NO!");
```

```
                    }}}
```

Output:



```
<terminated> setInterface [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Set 1: [Barley, WholeGrain, Lays, Wheat, Rice]
Set 2: [Biscuit, Oil, Cookies, Chocolate, Milk]
Set 3: [Ghee, Dettol, Coke]
Union of Set 2 & Set 3: [Ghee, Dettol, Biscuit, Oil, Cookies, Coke, Chocolate, Milk]
Set 3 when coke is removed: [Ghee, Dettol, Biscuit, Oil, Cookies, Chocolate, Milk]
What do You want to search in Set 1?
Barley
Question: Does my Set 1 has 'Barley' in it?
Ans: YES!
```



```
<terminated> setInterface [Java Application] C:\Users\Student\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Set 1: [Barley, WholeGrain, Lays, Wheat, Rice]
Set 2: [Biscuit, Oil, Cookies, Chocolate, Milk]
Set 3: [Ghee, Dettol, Coke]
Union of Set 2 & Set 3: [Ghee, Dettol, Biscuit, Oil, Cookies, Coke, Chocolate, Milk]
Set 3 when coke is removed: [Ghee, Dettol, Biscuit, Oil, Cookies, Chocolate, Milk]
What do You want to search in Set 1?
Onion
Question: Does my Set 1 has 'Onion' in it?
Ans: NO!
```

Q2) Program in Spring JDBC to demonstrate ResultSetExtractor Interface Book(bid,bname,bcategory)

pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sakshi</groupId>
  <artifactId>jdbc_1</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>jdbc_1</name>
  <url>http://maven.apache.org</url>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-jdbc</artifactId>
<version>5.3.5</version>
</dependency>
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.19</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>5.3.5</version>
</dependency>
  </dependencies>
</project>
```

Book.java

```
package com.sakshi.jdbc_1;

public class Book {
            private int bid; private String bname; private String category;
            public void setbId(int bid){
            this.bid=bid;
            }
            public int getbId(){
            return bid;
            }

            public void setbName(String name){
            this.bname=name;
            }
            public String getbName() {
            return bname;
            }

            public void setCategory(String category){
            this.category=category;
            }
            public String getCategory(){
            return category;
            }
            }
```

BookDAO.java

```
package com.sakshi.jdbc_1;
import java.util.List;
public interface BookDAO {
public List<Book> listBook();
}
```

BookTemplate.java

```
package com.sakshi.jdbc_1;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javax.sql.DataSource;
import org.springframework.dao.DataAccessException;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.ResultSetExtractor;
public class BookTemplate implements BookDAO {
private DataSource dataSource;
private JdbcTemplate jdbcTemplateObject;
public void setDataSource(DataSource dataSource) {
this.dataSource = dataSource;
this.jdbcTemplateObject = new JdbcTemplate(dataSource);
}
public List<Book>listBook(){
String url = "select * from Book";
return jdbcTemplateObject.query(url,new ResultSetExtractor<List<Book>>() {
public List<Book> extractData(ResultSet rs)throws SQLException,DataAccessException{
List<Book> ps = new ArrayList<Book>(); while(rs.next()) {
Book s = new Book();
s.setbId(rs.getInt(1));
s.setbName(rs.getString(2));
s.setCategory(rs.getString(3));
ps.add(s);
}
return ps;
}
});
}}
```

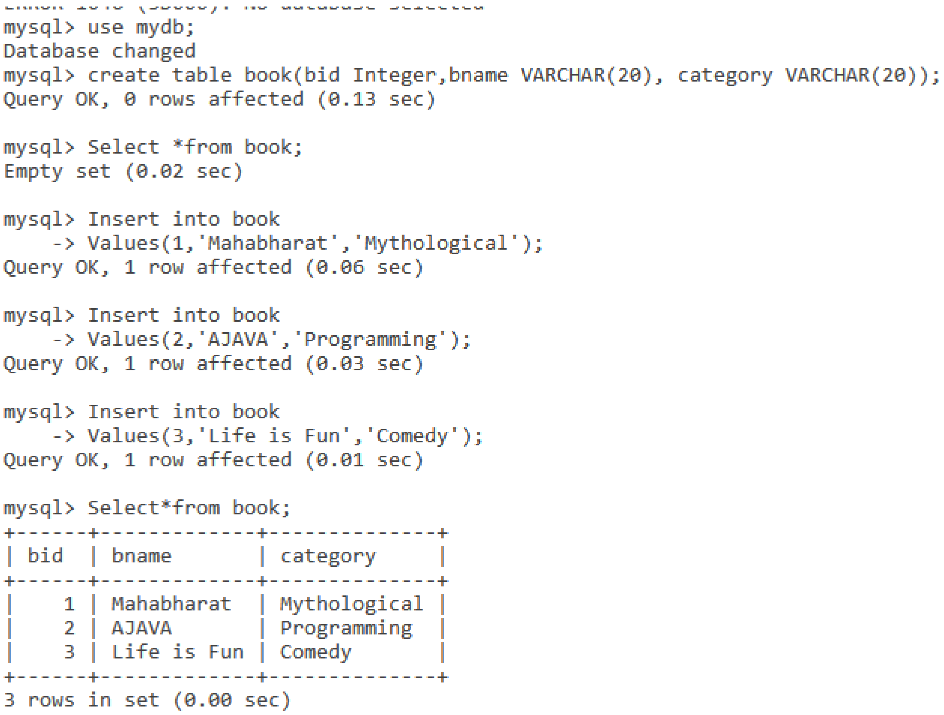
MainClass.java

```
package com.sakshi.jdbc_1;
import java.util.List;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class MainClass {
public static void main(String args[]) {
ApplicationContext context = new
ClassPathXmlApplicationContext("Bean.xml");
BookTemplate st = (BookTemplate) context.getBean("booktemplate");
//booktemplate is a bean id
List <Book> ps = st.listBook();
for(Book record:ps) {
System.out.println("Id="+record.getbId()+"Name="+record.getbName()+"Category="+record.get
Category());
}}}
```

Bean.xml

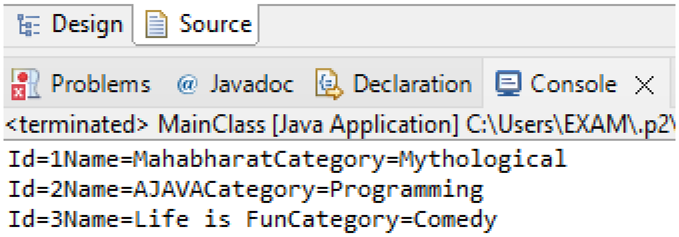
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd ">
<bean id="data_source"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="com.mysql.cj.jdbc.Driver"/>
<property name="url" value="jdbc:mysql://localhost:3306/mydb"/>
<property name="username" value="root"/>
<property name="password" value="root"/>
</bean>
<bean id="sportstemplate" class="com.sakshi.jdbc_1.BookTemplate">
<property name="dataSource" ref="data_source"/>
</bean>
</beans>
```

In MYSQL8.0 Command Line Client



```
ERROR 1046 (3D000): No database selected
mysql> use mydb;
Database changed
mysql> create table book(bid Integer,bname VARCHAR(20), category VARCHAR(20));
Query OK, 0 rows affected (0.13 sec)

mysql> Select *from book;
Empty set (0.02 sec)

mysql> Insert into book
    -> Values(1,'Mahabharat','Mythological');
Query OK, 1 row affected (0.06 sec)

mysql> Insert into book
    -> Values(2,'AJAVA','Programming');
Query OK, 1 row affected (0.03 sec)

mysql> Insert into book
    -> Values(3,'Life is Fun','Comedy');
Query OK, 1 row affected (0.01 sec)

mysql> Select*from book;
+------+-------------+--------------+
| bid  | bname       | category     |
+------+-------------+--------------+
|    1 | Mahabharat  | Mythological |
|    2 | AJAVA       | Programming  |
|    3 | Life is Fun | Comedy       |
+------+-------------+--------------+
3 rows in set (0.00 sec)
```

Output:



```
<terminated> MainClass [Java Application] C:\Users\EXAM\.p2\
Id=1Name=MahabharatCategory=Mythological
Id=2Name=AJAVACategory=Programming
Id=3Name=Life is FunCategory=Comedy
```