Student Name	Divyam Makhaik
UID	19BCS1928
Section &Group	19BCS-2
Department	Computer Science & Engineering
Session	Aug-Dec 2022
Course Name	WEB SERVICE SECURITY LAB
Course Code	CSB-422
Semester	7TH

Experiment: External Lab Practical

Aim:
Write a program to create a simple web service that calculate total money spend
in shopping.

Tools/Software Required:
Netbeans IDE 8.2, Java Development kit(JDK), Java runtime environment (JRE), GlassFish
Server,Web browser

Description:
WSDL stands for Web Service Description Language. It describes the web
services and exchanges the information in a decentralized or distributed environment. It is
used to create a relationship between the service requester and the service provider. It is an
XML based language to describe the functionality of a SOAP-based web service.

Pseudo code/Algorithms/Flowchart/Steps:

Creating a TemperatureApp Web Service
1.   Open Netbeans
2.   Open New Project → Java Web → Web Application.

Implementation:

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package General-Store;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.xml.ws.RequestWrapper;
import javax.xml.ws.ResponseWrapper;

/**
 *
 * @author lost
 */
/*

        getPrice



*/

@WebService(serviceName = "Buy_Product ")
public class Buy_Product {



    /**
     * Web service operation
     */
    @WebMethod(operationName = "buyItems")
    public Integer buyItems(@WebParam(name = "product1") String product1,
@WebParam(name = "product2") String product2, @WebParam(name = "product3") String
product3) {
        //TODO write your implementation code here:
        int total_money_spent  = 0 ;

//      product -1
```

NAME: Divyam Makhaik   UID: 19BCS1928

```
switch(product1){
    case "shirt" : {
        total_money_spent += 500;
        break;
    }
    case "pent" : {
        total_money_spent += 500;
        break;
    }
    case "H.P Laptop" : {
        total_money_spent += 50000;
        break;
    }
    case "Asus Laptop" : {
        total_money_spent += 100000;
        break;
    }
    case "charger" : {
        total_money_spent += 2000;
        break;
    }
}
//    product-2

    switch(product2){
        case "shirt" : {
            total_money_spent += 500;
            break;
        }
        case "pent" : {
            total_money_spent += 500;
            break;
        }
        case "H.P Laptop" : {
            total_money_spent += 50000;
            break;
        }
        case "Asus Laptop" : {
            total_money_spent += 100000;
            break;
        }
```

```
        case "charger" : {
            total_money_spent += 2000;
            break;
        }
    }
//      procuct -3
    switch(product3){
        case "shirt" : {
            total_money_spent += 500;
            break;
        }
        case "pent" : {
            total_money_spent += 500;
            break;
        }
        case "H.P Laptop" : {
            total_money_spent += 50000;
            break;
        }
        case "Asus Laptop" : {
            total_money_spent += 100000;
            break;
        }
        case "charger" : {
            total_money_spent += 2000;
            break;
        }
    }

    return  total_money_spent;
  }


}
```

Output:

← → 🌐 http://localhost:8080/WebApplication3/index?Tester ▾ ⭯ | Search... | 🔍▾ ⌂ ☆ ☼

🌐 index Web Service Tester ☒ ▢

index Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

---

public abstract java.lang.Integer newpack.Index.buyItems(java.lang.String,java.lang.String,java.lang.String)

[buyItems] ([Laptop] , [shirt] , [pent])

← ⊕ 🌐 http://localhost:8080/WebApplication3/index?Tester ▾ ⭯ | Search... | 🔍▾ ⌂ ☆ ☼ 🙂

🌐 Method invocation trace ☒ ▢

buyItems Method invocation

Method parameter(s)

Type	Value
java.lang.String	Laptop
java.lang.String	shirt
java.lang.String	pent

Method returned

java.lang.Integer : **"1000"**

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:buyItems xmlns:ns2="http://newPack/">
            <product1>Laptop</product1>
            <product2>shirt</product2>
            <product3>pent</product3>
        </ns2:buyItems>
    </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:buyItemsResponse xmlns:ns2="http://newPack/">
            <return>1000</return>
```

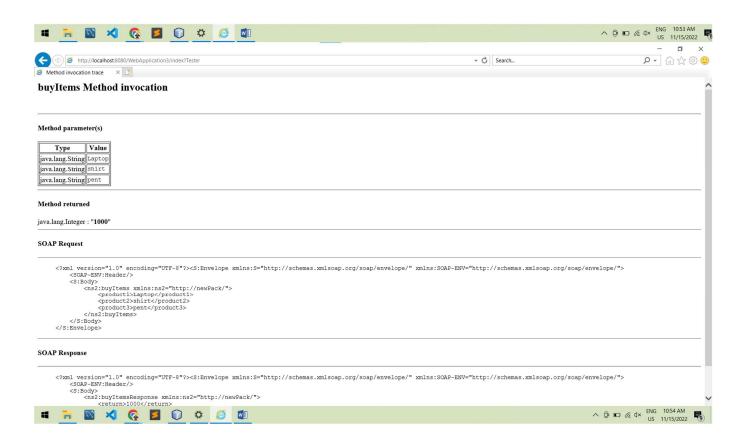NAME: Divyam Makhaik   UID: 19BCS1928

```
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:buyItemsResponse xmlns:ns2="http://newPack/">
            <return>1000</return>
        </ns2:buyItemsResponse>
    </S:Body>
</S:Envelope>
```

CourseName: WebServices Security Lab   **Course code:** CSB-422

ctoF Method invocation

Method parameter(s)

Type	Value
float	36.4

Method returned

float : **"97.520004"**

SOAP Request

```xml
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:CtoF xmlns:ns2="http://temperature.me.org/">
            <cel>36.4</cel>
        </ns2:CtoF>
    </S:Body>
</S:Envelope>
```

SOAP Response

```xml
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <S:Body>
        <ns2:CtoFResponse xmlns:ns2="http://temperature.me.org/">
            <return>97.520004</return>
        </ns2:CtoFResponse>
    </S:Body>
</S:Envelope>
```

NAME: Divyam Makhaik   UID: 19BCS1928