

```
1: // Grade.idl
2: // troy, 09/18/2K
3:
4: module Grade {
5:   interface Grader {
6:     boolean add_grade(in string tid, in string pwd, in float grade);
7:     float show_grade(in string sid, in string pwd);
8:   };
9:   interface Security {
10:    boolean check_teacher_pwd(in string tid, in string pwd);
11:    boolean check_student_pwd(in string sid, in string pwd);
12:  };
13: };
14:
```

```
1: // GraderImpl.java
2: // troy, 09/18/2K
3:
4: import org.omg.PortableServer.*;
5:
6: public class GraderImpl extends Grade.GraderPOA {
7:     private float _grade;
8:     private org.omg.CORBA.ORB _orb;
9:
10:
11:     public GraderImpl(org.omg.CORBA.ORB orb){
12:         _grade=(float)0.0;
13:         _orb=orb;
14:     }
15:
16:     public boolean add_grade(String tid, String pwd, float grade){
17:         boolean ret=false;
18:         byte[] securityId = "zzhan_security".getBytes();
19:         Grade.Security security =
20:             Grade.SecurityHelper.bind(_orb, "/zzhan_security_poa", securityId);
21:         boolean result = security.check_teacher_pwd(tid, pwd);
22:         if (!result)
23:             System.out.println("Grader(JAVA): Invalid TID or Password.");
24:         else{
25:             System.out.println("Grader(JAVA): Teacher "+tid+" logged in. Grade changed to: "+grade);
26:             _grade=grade;
27:             ret=true;
28:         }
29:         return ret;
30:     }
31:
32:     public float show_grade(String sid, String pwd){
33:         float ret=(float)-1.0;
34:         byte[] securityId = "zzhan_security".getBytes();
35:         Grade.Security security =
36:             Grade.SecurityHelper.bind(_orb, "/zzhan_security_poa", securityId);
37:         boolean result = security.check_student_pwd(sid, pwd);
38:         if (!result)
39:             System.out.println("Grader(JAVA): Invalid SID or Password.");
40:         else{
41:             System.out.println("Grader(JAVA): Student "+sid+" logged in. Grade "+_grade+" retrieved.");
42:             ret=_grade;
43:         }
44:         return ret;
45:     }
46: }
47:
```

```
1: // Grader.java
2: // troy, 09/19/2K
3:
4: import org.omg.PortableServer.*;
5:
6: public class Grader {
7:     public static void main(String[] args) {
8:         try {
9:             // Initialization.
10:            org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, null);
11:            POA rootPOA = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
12:            org.omg.CORBA.Policy[] policies = {
13:                rootPOA.create_lifespan_policy(LifespanPolicyValue.PERSISTENT)
14:            };
15:
16:            POA myPOA = rootPOA.create_POA( "zzhan_grader_poa", rootPOA.the_POAManager(), policies );
17:            GraderImpl graderServant = new GraderImpl(orb);
18:            byte[] graderId = "zzhan_Grader".getBytes();
19:            myPOA.activate_object_with_id(graderId, graderServant);
20:
21:            // Activate the POA manager
22:            rootPOA.the_POAManager().activate();
23:            System.out.println(myPOA.servant_to_reference(graderServant) +
24:                " is ready.");
25:
26:            // Wait for incoming requests
27:            orb.run();
28:        } catch (Exception e) {
29:            e.printStackTrace();
30:        }
31:    }
32:
33: }
34:
```

```
1: // SecurityImpl.java
2: // troy, 09/18/2K
3:
4: import org.omg.PortableServer.*;
5: import java.util.*;
6:
7: public class SecurityImpl extends Grade.SecurityPOA {
8:     private Dictionary _teacher_list = new Hashtable();
9:     private Dictionary _student_list = new Hashtable();
10:
11:     public SecurityImpl(){
12:         // Some built-in dummy entries
13:         _teacher_list.put("9999", "test");
14:         _teacher_list.put("8888", "test");
15:         _teacher_list.put("7777", "test");
16:
17:         _student_list.put("1001", "test");
18:         _student_list.put("2002", "test");
19:         _student_list.put("3003", "test");
20:     }
21:
22:     public synchronized boolean check_teacher_pwd(String tid, String pwd){
23:         boolean ret=false;
24:
25:         String stored_pwd = (String) _teacher_list.get(tid);
26:         if(stored_pwd==null){
27:             System.out.println("Security(JAVA): Created teacher "+ tid + " : " + pwd);
28:             _teacher_list.put(tid, pwd);
29:             ret=true;
30:         }else{
31:             if(pwd.equals(stored_pwd)){
32:                 System.out.println("Security(JAVA): Authentication succeeded on TID: "+tid);
33:                 ret=true;
34:             }else
35:                 System.out.println("Security(JAVA): Authentication failed on TID: "+tid);
36:         }
37:         return ret;
38:     }
39:
40:     public synchronized boolean check_student_pwd(String sid, String pwd){
41:         boolean ret=false;
42:         String stored_pwd = (String) _student_list.get(sid);
43:         if(stored_pwd==null){
44:             System.out.println("Security(JAVA): Created student "+ sid + " : " + pwd);
45:             _student_list.put(sid, pwd);
46:             ret=true;
47:         }else{
48:             if (pwd.equals(stored_pwd)){
49:                 System.out.println("Security(JAVA): Authentication succeeded on SID: "+sid);
```

```
50:         ret=true;
51:     }else
52:         System.out.println("Security(JAVA): Authentication failed on SID:"+sid);
53:     }
54:     return ret;
55: }
56: }
57: }
58: }
59: }
60: }
```

```
1: // Security.java
2: // troy, 09/18/2K
3:
4: import org.omg.PortableServer.*;
5: public class Security {
6:
7:     public static void main(String[] args) {
8:         try {
9:             // Initialization.
10:            org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, null);
11:            POA rootPOA = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
12:            org.omg.CORBA.Policy[] policies = {
13:                rootPOA.create_lifespan_policy(LifespanPolicyValue.PERSISTENT)
14:            };
15:            POA myPOA = rootPOA.create_POA( "zzhan_security_poa", rootPOA.the_POAManager(), policies );
16:
17:            SecurityImpl securityServant = new SecurityImpl();
18:            byte[] securityId = "zzhan_security".getBytes();
19:            myPOA.activate_object_with_id(securityId, securityServant);
20:
21:            // Activate the POA manager
22:            rootPOA.the_POAManager().activate();
23:            System.out.println(myPOA.servant_to_reference(securityServant) +
24:                " is ready.");
25:
26:            // Wait for incoming requests
27:            orb.run();
28:        }
29:        catch (Exception e) {
30:            e.printStackTrace();
31:        }
32:    }
33:
34: }
```

```
1: // Student.java
2: // troy, 09/18/2K
3:
4: public class Student {
5:     public static void main(String[] args) {
6:
7:         // Initialization
8:         org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, null);
9:         byte[] graderId = "zzhan_Grader".getBytes();
10:        Grade.Grader grader =
11:            Grade.GraderHelper.bind(orb, "/zzhan_grader_poa", graderId);
12:
13:        // Assuming args[0]: sid, args[1]: pwd
14:        String sid = args.length > 0 ? args[0] : "1001";
15:        String pwd = args.length > 1 ? args[1] : "test";
16:
17:        // Request the grader to show the grade.
18:        float grade = grader.show_grade(sid, pwd);
19:        if (grade < 0)
20:            System.out.println("Student(JAVA): Invalid SID or Password.");
21:        else
22:            System.out.println("Student(JAVA): The grade of "+sid+" is " + grade);
23:    }
24: }
25: }
```

```
1: // Teacher.java
2: // troy, 09/18/2K
3:
4: public class Teacher {
5:     public static void main(String[] args) {
6:
7:         // Initialization.
8:         org.omg.CORBA.ORB orb = org.omg.CORBA.ORB.init(args, null);
9:         byte[] graderId = "zzhan_Grader".getBytes();
10:        Grade.Grader grader =
11:            Grade.GraderHelper.bind(orb, "zzhan_grader_poa", graderId);
12:
13:        // Assume args[0]:tid, args[1]:pwd, args[2]:grade.
14:        String tid = args.length > 0 ? args[0] : "9999";
15:        String pwd = args.length > 1 ? args[1] : "test";
16:        float grade = (new Float(args.length > 2 ? args[2] : "75.50")).floatValue();
17:
18:        // Request the grader to add the grade.
19:        boolean result = grader.add_grade(tid, pwd, grade);
20:        if (result)
21:            System.out.println("Teacher(JAVA): Grade"+grade+" has been updated.");
22:        else
23:            System.out.println("Teacher(JAVA): Invalid TID or Password.");
24:    }
25:
26: }
```