# Regarding InternetBanking exam

1-      IDL: stubs

2-      Servant: implements

3-      Server: registers the object’s details as: name, address, phone, bank account…

4-      Client: invokes

 Client uses stub and send request to server and asks method, and server registers the objects details and returns method by skeleton to the client.

In online Internet Banking Exam:

We use Hash table for storing information instead of array-list, because hash table stores pairs (names and values), to return true or false.

**//in the below code, "ESBonLineDetails.UniqueID" is the name and "ESBonLineDetails" is value for name, value could be anything like: attributes, string...**

hashtable.put(ESBonLineDetails.UniqueID, ESBonLineDetails);

5- **Servant**

**The constructor for the server shall create and initialize online bank account for the ESB service provider. This account shall contain all the necessary details that are stored in the IDL struct such as name, address etc.**

//Setup the ESB a/c  
 //String timestamp=new Date().toString();  
 ESBID = "888";  
 onLineDetails ESBonLineDetails = new onLineDetails();  
 ESBonLineDetails.CustomerName = "ESB";  
 ESBonLineDetails.CustomerAddress = "Dublin";  
 ESBonLineDetails.Balance = 0;  
 ESBonLineDetails.BankACNumber=100;  
 ESBonLineDetails.UniqueID = ESBID;

 ID\_Counter = 1111;

Note: Also we use struct in these kinds of assignments instead of arrays, we used struct in: phone book, addressbook, online internetBanking...assignments to store data, because array can store just integers or strings but not both of them together.

It depends on what kinds (and how) of data we want to store, if we want to store real number, we use integer.

We can't convert name (string) to number (integer).

**// in the below code, "public void registerOnLineBanking" is the method and everything inside the brackets, are parameters (created files).**

Public void registerOnLineBanking (org.omg.CORBA.Any anyOnLineDetails, StringHolder password)

**// Register for on line banking ثبت مشخصات مشتریان، در بانک**

**The server shall create a unique ID using the counter described above. Note that the unique ID shall be of type string.**

**public void registerOnLineBanking (org.omg.CORBA.Any anyOnLineDetails, StringHolder password){  
 String timestamp=new Date().toString();  
// UniqueID = timestamp;  
 UniqueID = Integer.toString(++ID\_Counter);  
 onLineDetails myonLineDetails = onLineDetailsHelper.extract(anyOnLineDetails);  
 myonLineDetails.UniqueID = UniqueID;  
 hashtable.put(UniqueID, myonLineDetails);  
 password.value = UniqueID;**

**}**

**//** Perform an on line transaction (Pay ESB)

**The servant shall identify the correct details in the hash table using the unique ID that arrives in the service request.**

**public void performOnLineTransaction (String UniqueID, int amount, AnyHolder anyOnLineDetails){  
 try{  
  onLineDetails myonLineDetails = (onLineDetails)hashtable.get(UniqueID);  
  onLineDetails ESBonLineBanking = (onLineDetails) hashtable.get(ESBID);  
  myonLineDetails.Balance -= amount;  
  ESBonLineBanking.Balance += amount;  
  System.out.println("ESB Balance : "+ESBonLineBanking.Balance + ". Received " + amount + " from: "+myonLineDetails.CustomerName);**

**Any anyOnLine = orb.create\_any();  
  onLineDetailsHelper.insert(anyOnLine, myonLineDetails);  
  anyOnLineDetails.value = anyOnLine;**

**hashtable.put(UniqueID, myonLineDetails);  
  hashtable.put(ESBID, ESBonLineBanking);  
  }catch (Exception e) {System.err.println(e);  
  e.printStackTrace(System.out);  
  }**

**class in java:**

* **Anything we wish to represent in Java must be encapsulated in a class that defines the “state” and “behaviour” of the basic program components known as objects.**
* **Classes create objects and objects use methods to communicate between them. They provide a convenient method for packaging a group of logically related data items and functions that work on them**.
* A *class* is a collection of *fields* (data) and *methods* (procedure or function) that operate on that data.
* The fields (data) are also called the *instance* variables.
* Operations in Java are called methods

**یک کلاس، شامل ستیت یا همان حالت و رفتار یک آبجکت می باشد، مثلا فردی حالت عصبانیت دارد و در رفتار عصبانیتش را بگونه داد و بیداد و پرخاش کردن به دیگران و پرت کردن چیزی نشان میدهد.**

Object is value and variables are containers that stores and holds the values or objects and data types (INT or string).

Usually we don’t change server’s codes, we just writing and adding codes on servant, because server rplies client’s request by servant and returns method.