Java 1.8 This documentation is for reference purpose only and is for those who have attended the classroom sessions at Thinking Machines.

• During your classroom session appropriate theory needs to be written against each example.

• You are required to bring this book daily for your classroom sessions.

• Some examples won't compile. They have been written to explain some rules.

• If you try to understand the examples without attending theory sessions then may god help you.

Designing an event notification system

eg1.java (will compile)

```
class Bulb
{
private int wattage;
private WattageChangedLogger wattageChangedLogger;
public void setWattageChangedLogger(WattageChangedLogger wattageChangedLogger)
{
this.wattageChangedLogger=wattageChangedLogger;
}
public void setWattage(int wattage)
{
if(wattage!=this.wattage)
{
int oldWattage=this.wattage;
this.wattage=wattage;
if(this.wattageChangedLogger!=null)
{
wattageChangedLogger.wattageChanged(oldWattage,this.wattage);
}
}
}
public int getWattage()
{
return this.wattage;
}
}
class WattageChangedLogger
{
public void wattageChanged(int oldWattage,int newWattage)
{
System.out.printf("Wattage change from %d to %d\n",oldWattage,newWattage);
}
}
class EG1App
{
public static void main(String kk[])
{
Bulb b1=new Bulb();
WattageChangedLogger wcl=new WattageChangedLogger();
b1.setWattageChangedLogger(wcl);
b1.setWattage(60);
b1.setWattage(120);
b1.setWattage(120);
b1.setWattage(0);
}
}
```

Generalized event notification system & creating abstract class to impose guidelines.

eg2.java (will not compile)

```
// Assume that the following code is being written in year 2016
abstract class BulbEventListener
{
abstract public void wattageChanged(int oldWattage,int newWattage);
}
class Bulb
{
private int wattage;
private BulbEventListener bulbEventListener;
public void setBulbEventListener(BulbEventListener bulbEventListener)
{
this.bulbEventListener=bulbEventListener;
}
public void setWattage(int wattage)
{
if(wattage!=this.wattage)
{
int oldWattage=this.wattage;
this.wattage=wattage;
if(this.bulbEventListener!=null)
{
bulbEventListener.wattageChanged(oldWattage,this.wattage);
}
}
}
public int getWattage()
{
return this.wattage;
}
}
// Assume that the following code is being written after year 2016
class WattageChangedLogger
{
}
class EG2App
{
public static void main(String kk[])
{
Bulb b1=new Bulb();
WattageChangedLogger wcl=new WattageChangedLogger();
b1.setBulbEventListener(wcl);
b1.setWattage(60);
b1.setWattage(120);
b1.setWattage(120);
b1.setWattage(0);
}
}
```

eg2.java (will not compile)

```
// Assume that the following code is being written in year 2016
abstract class BulbEventListener
{
abstract public void wattageChanged(int oldWattage,int newWattage);
}
class Bulb
{
private int wattage;
private BulbEventListener bulbEventListener;
public void setBulbEventListener(BulbEventListener bulbEventListener)
{
this.bulbEventListener=bulbEventListener;
}
public void setWattage(int wattage)
{
if(wattage!=this.wattage)
{
int oldWattage=this.wattage;
this.wattage=wattage;
if(this.bulbEventListener!=null)
{
bulbEventListener.wattageChanged(oldWattage,this.wattage);
}
}
}
public int getWattage()
{
return this.wattage;
}
}
// Assume that the following code is being written after year 2016
class WattageChangedLogger extends BulbEventListener
{
}
class EG2App
{
public static void main(String kk[])
{
Bulb b1=new Bulb();
WattageChangedLogger wcl=new WattageChangedLogger();
b1.setBulbEventListener(wcl);
b1.setWattage(60);
b1.setWattage(120);
b1.setWattage(120);
b1.setWattage(0);
}
}
```

eg2.java (will compile)

```
// Assume that the following code is being written in year 2016
abstract class BulbEventListener
{
abstract public void wattageChanged(int oldWattage,int newWattage);
}
class Bulb
{
private int wattage;
private BulbEventListener bulbEventListener;
public void setBulbEventListener(BulbEventListener bulbEventListener)
{
this.bulbEventListener=bulbEventListener;
}
public void setWattage(int wattage)
{
if(wattage!=this.wattage)
{
int oldWattage=this.wattage;
this.wattage=wattage;
if(this.bulbEventListener!=null)
{
bulbEventListener.wattageChanged(oldWattage,this.wattage);
}
}
}
public int getWattage()
{
return this.wattage;
}
}
// Assume that the following code is being written after year 2016
class WattageChangedLogger extends BulbEventListener
{
public void wattageChanged(int oldWattage,int newWattage)
{
System.out.printf("Wattage change from %d to %d\n",oldWattage,newWattage);
}
}
class EG2App
```

```
{
public static void main(String kk[])
{
Bulb b1=new Bulb();
WattageChangedLogger wcl=new WattageChangedLogger();
b1.setBulbEventListener(wcl);
b1.setWattage(60);
b1.setWattage(120);
b1.setWattage(120);
b1.setWattage(0);
}
}
```

Problems associated with creating abstract class to impose guidelines

What if the programmer of WattageChangedLogger class wants to extend the WattageChangedLogger class from another class. He cannot do so as java doesn't support Multiple Inheritance. In such scenario extending an abstract class seems to be a burden which might not be acceptable in some scenarios. The creators of java introduced the feature of creating interface to impose guidelines.

Note : Interface is not an alternative to multiple inheritance. Interface in an alternative to an abstract class with no properties and whose all methods are abstract. The sole purpose of creating interface in to impose guidelines or provide declaration of a certain kind.

Interface

eg3.java (will not compile)

```
interface aaaa
{
private void sam() // wrong
{
}
public void tom() // wrong
{
}
public void john(); // correct
}
```

---

eg4.java (will not compile)

```
interface aaaa
{
public void john(); // correct
}
class bbb extends aaaa   // incorrect
{

}
```

---

eg5.java (will not compile)

```
interface aaaa
{
public void john(); // correct
}
class bbb implements aaaa // wrong
{
}
abstract class ccc implements aaaa // correct
{

}
class ddd implements aaaa // correct
{
public void john()
{
// some code or whatever is required
}
public void tom()
{
// whatever
}
}
```

---

eg6.java (will not compile)

```
interface aaaa
{
public void john(); // correct
}
class psp
{
public static void main(String gg[])
{
aaaa a; // correct
a=new aaaa(); // incorrect
}
}
```

---

eg7.java (will not compile)

```
interface aaaa
{
public void john(); // correct
}
class bbb implements aaaa // correct
{
public void john()
{
// some code or whatever is required
}
public void tom()
{
// whatever
}
}
class ccc extends bbb
{
// some functions
}
class ddd
{
public void john()
{
// some code
}
}
class psp
{
public static void main(String gg[])
{
aaaa a; // correct
a=new bbb(); // correct
a.john(); // correct
a.tom(); // incorrect
a=new ccc(); // correct
a=new ddd(); // incorrect
}
}
```

Some more cases

Assume that the implemented interfaces and extended classes exist

```
class jjjj extends pqr implements aaaa  // correct

class jjjj implements aaaa extends pqr // incorrect

class jjjj implements aaaa,bbbb,ccccc,ddddd // correct
```

---

eg8.java (will compile)

```
interface aaaa
{
void sam(); // correct, compiler will declare it as public
}
```

---

eg9.java (will not compile)

```
interface aaaa
{
public int x;
}
```

---

eg10.java (will not compile)

```
interface aaaa
{
public int x=10; // a variable declared in an interface will become final
}
class bbb implements aaaa
{
public void tom()
{
x=20; // wrong
x=10; // wrong
}
}
```

---

eg11.java (will not compile)

```
interface aaaa
{
public void sam()
{
System.out.println("Great");
}
}
```

---

eg12.java (will compile)

```
interface aaaa
{
default public void sam()
{
System.out.println("Great");
}
}class bbb implements aaaa
{
public void tom()
{
System.out.println("Cool");
}
}
class EG3App
{
public static void main(String kk[])
{
aaaa a=new bbb();
a.sam();
}
}
```

---

eg13.java (will compile)

```
interface aaaa
{
default public void sam()
{
System.out.println("Great");
}
}class bbb implements aaaa
{
public void sam()
{
System.out.println("Super cool");
}
public void tom()
{
System.out.println("Cool");
}
}
class EG4App
{
public static void main(String kk[])
{
aaaa a=new bbb();
a.sam();
}
}
```

Keyboard input

eg15.java (will not compile)

```
import java.io.*;
class EG5App
{
public static void main(String kk[])
{
InputStreamReader isr;
isr=new InputStreamReader(System.in);
BufferedReader br;
br=new BufferedReader(isr);
char m;
System.out.print("Enter a character : ");
m=(char)br.read();
System.out.println(m);
}
}
```

---

eg15.java (will compile)

```
import java.io.*;
class EG5App
{
public static void main(String kk[])
{
InputStreamReader isr;
isr=new InputStreamReader(System.in);
BufferedReader br;
br=new BufferedReader(isr);
char m;
System.out.print("Enter a character : ");
try
{
m=(char)br.read();
System.out.println(m);
}catch(IOException ioException)
{
System.out.println(ioException);
}
}
}
```

eg16.java (will compile)

```
import java.io.*;
class EG6App
{
public static void main(String kk[])
{
InputStreamReader isr;
isr=new InputStreamReader(System.in);
BufferedReader br;
br=new BufferedReader(isr);
char m;
System.out.print("Enter a character : ");
try
{
m=(char)br.read();
System.out.println(m);
}catch(IOException ioException)
{
System.out.println(ioException);
}
char t;
System.out.print("Enter another character : ");
try
{
t=(char)br.read();
}catch(IOException ioException)
{
System.out.println(ioException);
}
}
}
```

eg17.java (will compile)

```
import java.io.*;
class EG7App
```

```
{
public static void main(String kk[])
{
InputStreamReader isr;
isr=new InputStreamReader(System.in);
BufferedReader br;
br=new BufferedReader(isr);
char m;
System.out.print("Enter a character : ");
try
{
m=(char)br.read();
while(br.ready()) br.read();
System.out.println(m);
}catch(IOException ioException)
{
System.out.println(ioException);
}
char t;
System.out.print("Enter another character : ");
try
{
t=(char)br.read();
while(br.ready()) br.read();
}catch(IOException ioException)
{
System.out.println(ioException);
}
String k;
System.out.print("Enter a string : ");
try
{
k=br.readLine();
System.out.println(k);
}catch(IOException ioException)
{
System.out.println(ioException);
}
}
}
```

Generalized Keyboard class
eg18.java (will compile)

```
import java.io.*;
class Keyboard
{
private static BufferedReader bufferedReader=new BufferedReader(new
InputStreamReader(System.in));
private Keyboard()
{
}
public static char readCharacter()
{
char m=' ';
try
{
m=(char)bufferedReader.read();
while(bufferedReader.ready())
{
bufferedReader.read();
}
}catch(IOException ioException)
{
}
return m;
}
public static char readCharacter(String message)
{
System.out.print(message);
return readCharacter();
}
public static int readInteger()
{
return Integer.parseInt(readString());
}
public static int readInteger(String message)
{
System.out.print(message);
return readInteger();
}
public static double readDouble()
{
return Double.parseDouble(readString());
}
public static double readDouble(String message)
{
System.out.print(message);
return readDouble();

}
public static String readString()
{
String m="";
try
```

```
{
m=bufferedReader.readLine();
}catch(IOException ioException) {}
return m;
}
public static String readString(String message)
{
System.out.print(message);
return readString();
}
/* write implementations for
readLong()
readShort()
readByte()
readDouble()
readFloat()
readBoolean()
overload all the above to accept a string as message
*/
}
class EG8App
{
public static void main(String gg[])
{
char a;
a=Keyboard.readCharacter("Enter a character : ");
System.out.println(a);
char b=Keyboard.readCharacter("Enter another character : ");
System.out.println(b);
int x;
x=Keyboard.readInteger("Enter a number : ");
System.out.print("Enter another number : ");
int y;
y=Keyboard.readInteger();
int z=x+y;
System.out.printf("Total is %d\n",z);
String g;
g=Keyboard.readString("Enter a string :");
System.out.println(g);
}
}
```

Predefined Scanner class
eg19.java (will compile)

```
import java.util.*;
class EG9App
{
```

```
public static void main(String kk[])
{
Scanner scanner=new Scanner(System.in);
System.out.print("Enter name : ");
String name=scanner.nextLine();
System.out.println(name);
System.out.print("Enter age : ");
int age=scanner.nextInt();
System.out.println(age);
System.out.print("Enter gender (M/F) : ");
char gender=(char)scanner.next().charAt(0);
System.out.println(gender);
}
}
```

File Handling (CRUD Operations)

eg20.java (will compile)

```
import java.io.*;
class AddEmployee
{
public static void main(String data[])
{
if(data.length!=3)
{
System.out.println("Invalid use of module : AddEmployee");
System.out.println("Usage : java AddEmployee code name salary");
return;
}
int code=Integer.parseInt(data[0]);
String name=data[1];
int salary=Integer.parseInt(data[2]);
try
{
File f;
f=new File("employee.data");
// because of the above code, don't assume that the file has been opened
RandomAccessFile raf;
/*
because of the following line, the constructor of the
RandomAccessFile class will open a file in RAM and some
internal pointer will point to the first byte of the file.
If the file doesn't exist, a new file will be opened, we can write/read
because of the mode (rw) another available mode is (r)
*/
raf=new RandomAccessFile(f,"rw");
/*
length() function returns the length of the file
getFilePointerFunction() returns the current position
of the pointer (First is looked upon as zero)
readLine() function returns a string(reads till \n is found)
writeBytes(String) will write the string from current position
*/
int vCode;
String vName;
int vSalary;
while(raf.getFilePointer()<raf.length())
{
vCode=Integer.parseInt(raf.readLine());
vName=raf.readLine();
vSalary=Integer.parseInt(raf.readLine());
if(vCode==code)
{
raf.close();
System.out.println("That code alloted to : "+vName);
return;
}
}
// valueOf to convert anything to String
raf.writeBytes(String.valueOf(code));
raf.writeBytes("\n");
raf.writeBytes(name);
raf.writeBytes("\n");
raf.writeBytes(String.valueOf(salary));
raf.writeBytes("\n");
raf.close();
System.out.println("Employee added.......");
}catch(IOException ioException)
{
System.out.println("Problem : "+ioException.getMessage());
}
}
}
```

eg21.java (will compile)

```
import java.io.*;
class UpdateEmployee
{
public static void main(String data[])
{
if(data.length!=3)
{
System.out.println("Invalid use of module : UpdateEmployee");
System.out.println("Usage : java UpdateEmployee code name salary");
return;
```

```
}
int code=Integer.parseInt(data[0]);
String name=data[1];
int salary=Integer.parseInt(data[2]);
try
{
File f=new File("employee.data");
if(f.exists()==false)
{
System.out.println("Invalid code");
return;
}
RandomAccessFile raf;
raf=new RandomAccessFile(f,"rw");
if(raf.length()==0)
{
System.out.println("Invalid code");
raf.close();
return;
}
int vCode;
String vName;
int vSalary;
boolean found=false;
while(raf.getFilePointer()<raf.length())
{
vCode=Integer.parseInt(raf.readLine());
vName=raf.readLine();
vSalary=Integer.parseInt(raf.readLine());
if(vCode==code)
{
found=true;
break;
}
}
if(found==false)
{
raf.close();
System.out.println("Invalid code");
return;
}
raf.seek(0); // seek will move the internal file pointer to desired location
File tmpFile=new File("tmp.data");
if(tmpFile.exists())
{
tmpFile.delete();
}
```

```
RandomAccessFile tmpraf=new RandomAccessFile(tmpFile,"rw");
while(raf.getFilePointer()<raf.length())
{
vCode=Integer.parseInt(raf.readLine());
vName=raf.readLine();
vSalary=Integer.parseInt(raf.readLine());
if(code!=vCode)
{
tmpraf.writeBytes(vCode+"\n"+vName+"\n"+vSalary+"\n");
}
else
{
tmpraf.writeBytes(code+"\n"+name+"\n"+salary+"\n");
}
}
raf.seek(0);
tmpraf.seek(0);
while(tmpraf.getFilePointer()<tmpraf.length())
{
raf.writeBytes(tmpraf.readLine()+"\n");
}
raf.setLength(tmpraf.length());
tmpraf.setLength(0);
raf.close();
tmpraf.close();
System.out.println("Employee updated.....");
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
```

eg22.java (will compile)

```
import java.io.*;
class DeleteEmployee
{
public static void main(String data[])
{
if(data.length!=1)
{
System.out.println("Invalid use of module : DeleteEmployee");
System.out.println("Usage : java DeleteEmployee code");
return;
}
int code=Integer.parseInt(data[0]);
try
```

```
{
File f=new File("employee.data");
if(f.exists()==false)
{
System.out.println("Invalid code");
return;
}
RandomAccessFile raf;
raf=new RandomAccessFile(f,"rw");
if(raf.length()==0)
{
System.out.println("Invalid code");
raf.close();
return;
}
int vCode;
String vName;
int vSalary;
boolean found=false;
while(raf.getFilePointer()<raf.length())
{
vCode=Integer.parseInt(raf.readLine());
vName=raf.readLine();
vSalary=Integer.parseInt(raf.readLine());
if(vCode==code)
{
found=true;
break;
}
}
if(found==false)
{
raf.close();
System.out.println("Invalid code");
return;
}
raf.seek(0); // seek will move the internal file pointer to desired location
File tmpFile=new File("tmp.data");
if(tmpFile.exists())
{
tmpFile.delete();
}
RandomAccessFile tmpraf=new RandomAccessFile(tmpFile,"rw");
while(raf.getFilePointer()<raf.length())
{
vCode=Integer.parseInt(raf.readLine());
vName=raf.readLine();
vSalary=Integer.parseInt(raf.readLine());
if(code!=vCode)
{
tmpraf.writeBytes(vCode+"\n"+vName+"\n"+vSalary+"\n");
}
}
raf.seek(0);
tmpraf.seek(0);
while(tmpraf.getFilePointer()<tmpraf.length())
{
raf.writeBytes(tmpraf.readLine()+"\n");
}
raf.setLength(tmpraf.length());
tmpraf.setLength(0);
raf.close();
tmpraf.close();
System.out.println("Employee deleted.....");
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
```

eg23.java (will compile)

```
import java.io.*;
class GetEmployee
{
public static void main(String data[])
{
if(data.length!=1)
{
System.out.println("Invalid use of module : GetEmployee");
System.out.println("Usage : java GetEmployee code");
return;
}
int code=Integer.parseInt(data[0]);
try
{
File f;
f=new File("employee.data");
if(f.exists()==false)
{
System.out.println("Invalid employee code");
return;
}
RandomAccessFile raf;
raf=new RandomAccessFile(f,"rw");
if(raf.length()==0)
{
raf.close();
System.out.println("Invalid employee code");
return;
}
int vCode;
String vName;
int vSalary;
while(raf.getFilePointer()<raf.length())
{
vCode=Integer.parseInt(raf.readLine());
vName=raf.readLine();
vSalary=Integer.parseInt(raf.readLine());
if(vCode==code)
{
raf.close();
System.out.println("Name : "+vName);
System.out.println("Salary : "+vSalary);
return;
}
}
raf.close();
System.out.println("Invalid employee code");
}catch(IOException ioException)
{
System.out.println("Problem : "+ioException.getMessage());
}
}
}
```

RDBMS (SQLite)

Download sqlite.jar and sqlite3.zip

Unzip the contents from sqlite3.zip and copy the contents to c:\sqlite3 folder.
Copy the sqlite.jar to c:\sqlite3 folder

Add c:\sqlite3 to PATH environment variable as you add c:\jdk1.8\bin etc.

set PATH=c:\windows;c:\windows\system32;c:\jdk1.8\bin;c:\sqlite3
create a folder named as sqleg on c:\

Now while staying in c:\sqleg folder, create a database using the following statement

```
sqlite3    stock.db
```

sqlite prompt will appear, type the following sql statements and terminate them with ;

Note down all the results in your copy, even if it takes a long time to do that. It is necessary to understand sql statements once and for all.

```
Create table item
(
code integer primary key,
name text,
unit_of_measurement
);

insert into item values(101,'Screw','Nos');
insert into item values(102,'Computer','NOS');
insert into item values(103,'Milk','Ltr');
insert into item values(102,'Printer','Nos')

select * from item

.headers on

select * from item

select code,name from item
select name,code from item
select name,code,name from item
select code,name as "Name" from item

update item set unit_of_measurement='Packet' where code=101
select * from item

update item set name='curd',unit_of_measurement='Kg' where code=103
select * from item
delete from item where code=102
select * from item
delete from item
select * from item
```

now type (.quit) to exit from sqlite

This page has been intentionally left blank for SQL Statements.

This page has been intentionally left blank for SQL Statements.

JDBC

jdbc1.java (will compile)

```
import java.sql.*;
class jdbc1
{
public static void main(String gg[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
s.executeUpdate("insert into item values(101,'Screw','Nos')");
s.close();
c.close();
System.out.println("Record added");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

---

compile the above code using (javac jdbc1.java)

to run type (java   -classpath   c:\sqlite3\sqlite.jar;.   Jdbc1)

You should see a message, record added.

Run again and now you should see an exception.

Now type (sqlite3   stock.db)

type the following on sqlite prompt

select * from item;

You should see the record that we added from a java code

Now create the following java file to update record

jdbc2.java (will compile)

```
import java.sql.*;
class jdbc2
{
public static void main(String gg[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
s.executeUpdate("update item set name='Computer',unit_of_measurement='Packet' where code=101");
s.close();
c.close();
System.out.println("Record updated");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

to compile (javac jdbc2.java)

to run type (java -classpath c:\sqlite3\sqlite.jar;. Jdbc2

you should see a message (record updated) now go into sqlite3 and check if the record has been updated or not.

Type the following code to delete a record

jdbc3.java (will compile)

```
import java.sql.*;
class jdbc3
{
public static void main(String gg[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
s.executeUpdate("delete from item where code=101");
s.close();
c.close();
System.out.println("Record deleted");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

Compile and run the above program as done earlier and verify using sqlite3

Now let us make the values dynamic by accepting them as command line arguments

jdbc4.java (will compile)

```
import java.sql.*;
class jdbc4
{
public static void main(String data[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
s.executeUpdate("insert into item values("+data[0]+",'"+data[1]+"','"+data[2]+"')");
s.close();
c.close();
System.out.println("Record added");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

---

compile the above code using (javac jdbc4.java)

for execution

```
java  -classpath  c:\sqlite3\sqlite;.jar   jdbc4   101   Screw   Packet
```

add some more records.

Now let us write a code for updation that accepts data as command line arguments

jdbc5.java (will compile)

```
import java.sql.*;
class jdbc5
{
public static void main(String data[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
s.executeUpdate("update item set name='"+data[1]+"',unit_of_measurement='"+data[2]+"' where
code="+data[0]);
s.close();
c.close();
System.out.println("Record updated");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

compile and run as learnt earlier

jdbc6.java (will compile)

```
import java.sql.*;
class jdbc6
{
public static void main(String data[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
s.executeUpdate("delete from item where code="+data[0]);
s.close();
c.close();
System.out.println("Record deleted");
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

jdbc7.java (will compile)

```
import java.sql.*;
class jdbc7
{
public static void main(String data[])
{
try
{
Class.forName("org.sqlite.JDBC");
Connection c;
c=DriverManager.getConnection("jdbc:sqlite:stock.db");
Statement s=c.createStatement();
ResultSet r;
r=s.executeQuery("select * from item");
int vCode;
String vName;
String vUOM;
while(r.next())
{
vCode=r.getInt("code");
vName=r.getString("name").trim();
vUOM=r.getString("unit_of_measurement").trim();
System.out.println(vCode+","+vName+","+vUOM);
}
r.close();
s.close();
c.close();
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

Now let us learn joins

create the following 3 tables (country,state and city using the following SQL statements)

```
create table country
(
code integer primary key,
name text unique
)

create table state
(
code integer primary key,
name text
country_code integer
)

create table city
(
code integer primary key,
name text,
state_code integer
)
```

following is the SQLite dump for sql statement, understand the basics yourself

```
C:\sqleg>sqlite3   stock.db
SQLite version 3.7.15.1 2012-12-19 20:39:10
```

```
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> create table country
   ...> (code integer primary key,name text unique);
sqlite> create table state
   ...> (code integer primary key,name text,country_code integer);
sqlite> create table city
   ...> (code integer primary key,name text,state_code integer);
sqlite> insert into country values(1,'India');
sqlite> insert into country values(2,'Pakistan');
sqlite> insert into state values(1,'M.P.',1);
sqlite> insert into state values(2,'U.P.',1);
sqlite> insert into state values(3,'Maharashtra',1);
sqlite> insert into state values(4,'Punjab',2);
sqlite> select * from country;
1|India
2|Pakistan
sqlite> select * from state;
1|M.P.|1
2|U.P.|1
3|Maharashtra|1
4|Punjab|2
sqlite> .headers on
sqlite> select * from state,country
   ...> ;
code|name|country_code|code|name
1|M.P.|1|1|India
2|U.P.|1|1|India
3|Maharashtra|1|1|India
4|Punjab|2|1|India
1|M.P.|1|2|Pakistan
2|U.P.|1|2|Pakistan
3|Maharashtra|1|2|Pakistan
4|Punjab|2|2|Pakistan
sqlite> select * from state,country where state.country_code=country.code;
code|name|country_code|code|name
1|M.P.|1|1|India
2|U.P.|1|1|India
3|Maharashtra|1|1|India
4|Punjab|2|2|Pakistan
sqlite> select code,name,code,name from state,country where state.country_code=c
ountry.code;
Error: ambiguous column name: code
sqlite> select state.code,state.name,country.code,country.name from state,countr
y where state.country_code=country.code;
code|name|code|name
1|M.P.|1|India
```

```
2|U.P.|1|India
3|Maharashtra|1|India
4|Punjab|2|Pakistan
sqlite> select city.name as "city",state.name as "state",country.name as "countr
y" from city,state,country
   ...> ;
sqlite> insert into city values(1,'Ujjain',1);
sqlite> insert into city values(2,'Indore',1);
sqlite> insert into city values(3,'Mumbai',3);
sqlite> insert into city values(4,'Pune',3);
sqlite> insert into city values(5,'Satara',3);
sqlite> select city.name as "city",state.name as "state",country.name as "countr
y" from city,state,country
   ...> ;
city|state|country
Ujjain|M.P.|India
Ujjain|U.P.|India
Ujjain|Maharashtra|India
Ujjain|Punjab|India
Indore|M.P.|India
Indore|U.P.|India
Indore|Maharashtra|India
Indore|Punjab|India
Mumbai|M.P.|India
Mumbai|U.P.|India
Mumbai|Maharashtra|India
Mumbai|Punjab|India
Pune|M.P.|India
Pune|U.P.|India
Pune|Maharashtra|India
Pune|Punjab|India
Satara|M.P.|India
Satara|U.P.|India
Satara|Maharashtra|India
Satara|Punjab|India
Ujjain|M.P.|Pakistan
Ujjain|U.P.|Pakistan
Ujjain|Maharashtra|Pakistan
Ujjain|Punjab|Pakistan
Indore|M.P.|Pakistan
Indore|U.P.|Pakistan
Indore|Maharashtra|Pakistan
Indore|Punjab|Pakistan
Mumbai|M.P.|Pakistan
Mumbai|U.P.|Pakistan
Mumbai|Maharashtra|Pakistan
Mumbai|Punjab|Pakistan
```

```
Pune|M.P.|Pakistan
Pune|U.P.|Pakistan
Pune|Maharashtra|Pakistan
Pune|Punjab|Pakistan
Satara|M.P.|Pakistan
Satara|U.P.|Pakistan
Satara|Maharashtra|Pakistan
Satara|Punjab|Pakistan
sqlite> select city.name as "city",state.name as "state",country.name as "countr
y" from city,state,country where city.state_code=state.code and state.country_co
de=country.code;
city|state|country
Ujjain|M.P.|India
Indore|M.P.|India
Mumbai|Maharashtra|India
Pune|Maharashtra|India
Satara|Maharashtra|India
sqlite>
sqlite> select count(*) from city;
5
sqlite> select count(*) from city where state_code=3;
3
sqlite> select count(*) from city where state_code=(select code from state where
 name='M.P.');
2
sqlite> select name from state where code not in (select state_code from city);
U.P.
Punjab
sqlite>
```

MySQL

Install MySql.
While installing don't change the default port number (3306) and for root user assign your surname as root password.

Now create a folder named as mysqleg on c:\

Locate the installation folder of MySQL Server. (It must be under Program Files or Program Files (x86) folder, in the MySQL Server installation folder there must be a bin folder, in the bin folder resides the mysql.exe (the Command Line Interface client tool to connect to the MySQL Server).

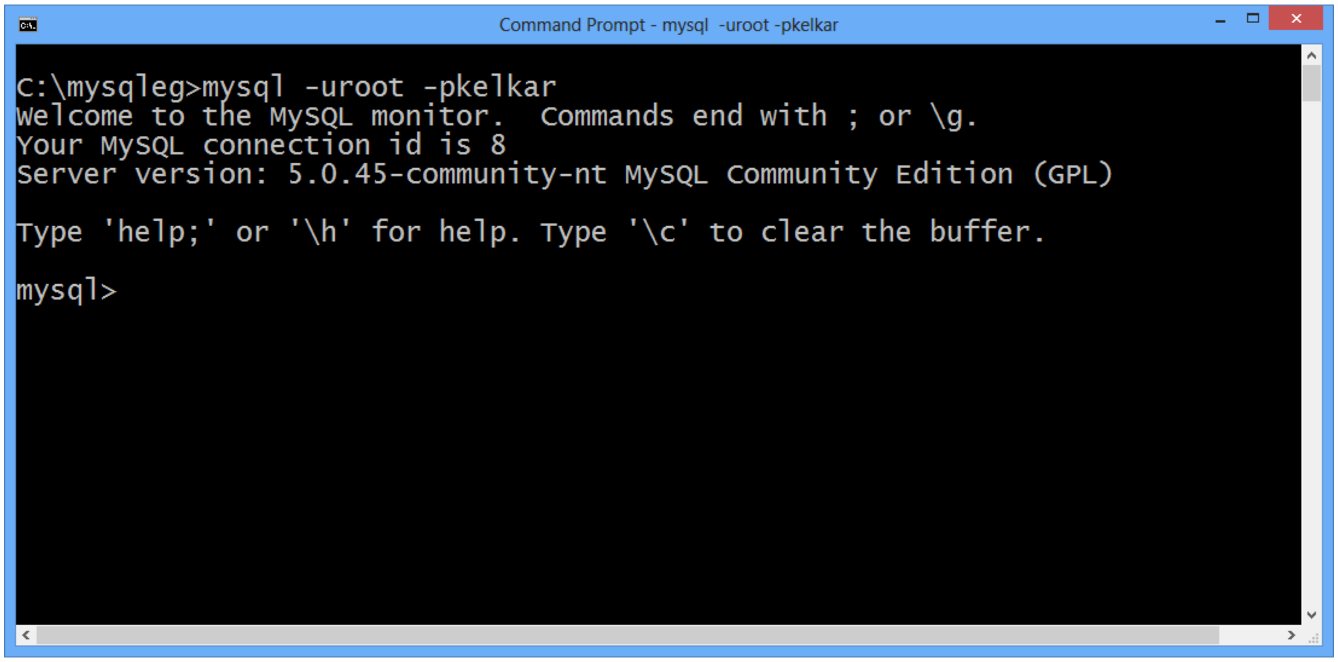
Now copy the mysql.exe to c:\mysqleg (or add the path upto the bin folder that contains the mysql.exe to the PATH environment variable)

Now move to c:\mysqleg folder and type

mysql -uroot -pkelkar

Note : replace kelkar with whatever is your root password. Also note that there is not space after -u and -p.

If everything is correct, you should see the mysql prompt as shown below.



```
C:\mysqleg>mysql -uroot -pkelkar
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

This is where we will be typing our SQL Statements.

Note : End all the SQL Statements with a semicolon (;). Till you don't type semicolon, all the statements are stored in buffer by mysql and when the semicolon is provided in the end, the collected statement is fired.

First of all we will create a database named as ThinkingMachinesDB
for that type

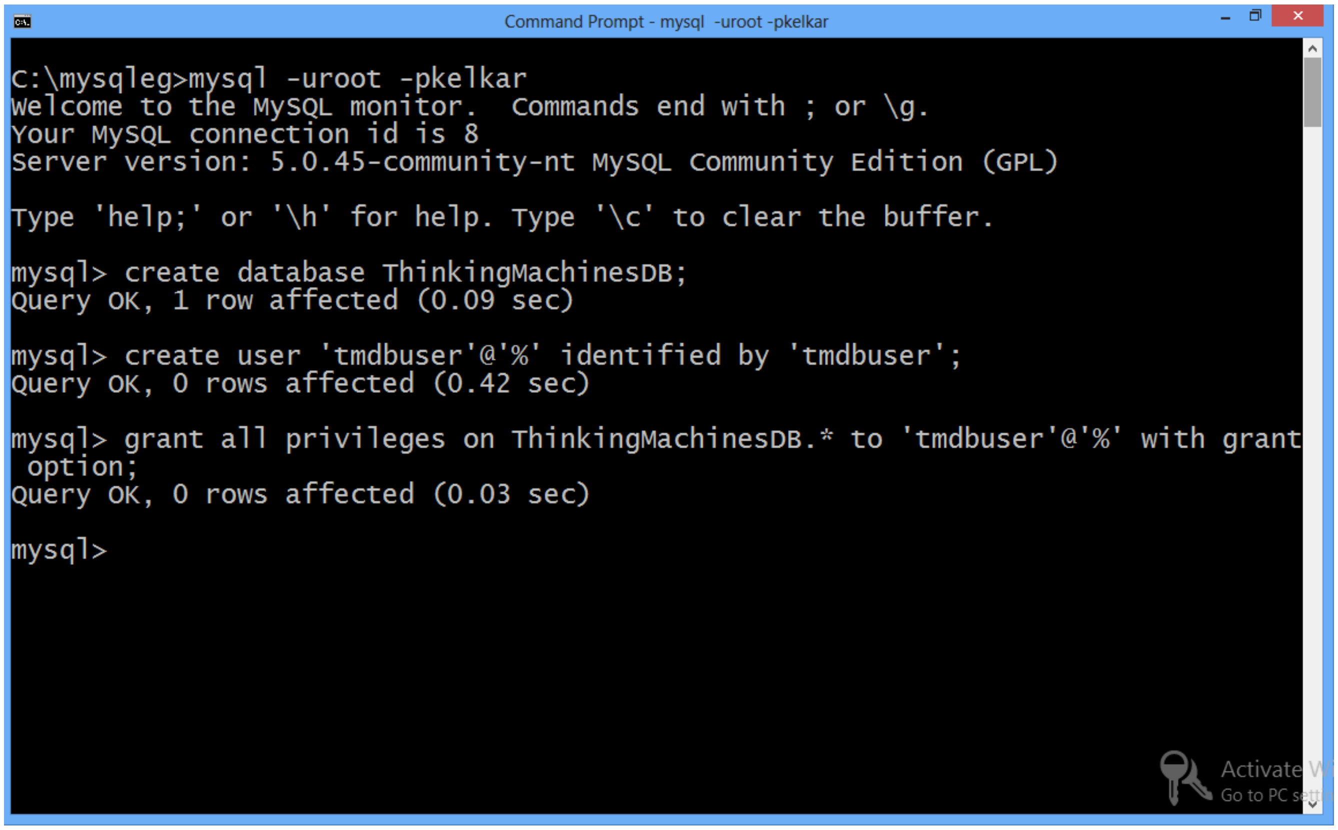
create database ThinkingMachinesDB;

Now let us create a user account named as tmdbuser with password also as tmdbuser, for that type

create user 'tmdbuser'@'%' identified by 'tmdbuser';

Now let us grant all the rights of the ThinkingMachinesDB to tmdbuser, for that type

grant all privileges on ThinkingMachinesDB.* to 'tmdbuser'@'%' with grant option;



```
C:\mysqleg>mysql -uroot -pkelkar
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database ThinkingMachinesDB;
Query OK, 1 row affected (0.09 sec)

mysql> create user 'tmdbuser'@'%' identified by 'tmdbuser';
Query OK, 0 rows affected (0.42 sec)

mysql> grant all privileges on ThinkingMachinesDB.* to 'tmdbuser'@'%' with grant
 option;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Now type quit to exit from MySQL

Now again login, but this time as tmdbuser, for that type

mysql -utmdbuser -ptmdbuser

If everything was done properly, you should see the mysql prompt, now to select the database type
use ThinkingMachinesDB

you should see the message that says Database Changed.

Now create tables using the following sql statements

```
create table item
(code int primary key auto_increment,
name char(35) not null unique,
opening_stock int not null default 0 ,
total_purchases int not null default 0 ,
total_sales int not null default 0 ,
closing_stock int not null default 0 ) Engine=InnoDB;
```

```
create table customer
(code int primary key auto_increment,
name char(50) not null unique)Engine=InnoDB;
```

```
create table supplier
(code int primary key auto_increment,
name char(50) not null unique)Engine=InnoDB;
```

```
create table sale
(bill_number int primary key auto_increment,
bill_date date not null,
customer_code int not null references customer)Engine=InnoDB;
```

```
create table sale_item
(bill_number int not null references sale,
item_code int not null references item,
quantity int not null ,
rate double not null ,
primary key(bill_number,item_code))Engine=InnoDB;
```

```
create table purchase
(reference_number int primary key auto_increment,
bill_number char(25) not null,
bill_date date not null,
supplier_code int not null references supplier)Engine=InnoDB;
```

```
create table purchase_item
(reference_number int not null references purchase,
item_code int not null references item,
quantity int not null ,
rate double not null ,
primary key(reference_number,item_code))Engine=InnoDB;
```

Note : you can create individual sql statement in a separate file using any plain text editor (for example item.sql which contains the sql statement to create item table) and then on mysql prompt you can type

```
source  item.sql
```

To get list of tables, you can type

```
show   tables;
```

To get details of each table, you can type
describe tablename;

Now let us create triggers to update the item table whenever insert/update/delete operations are performed on sale_item and purchase_item tables.

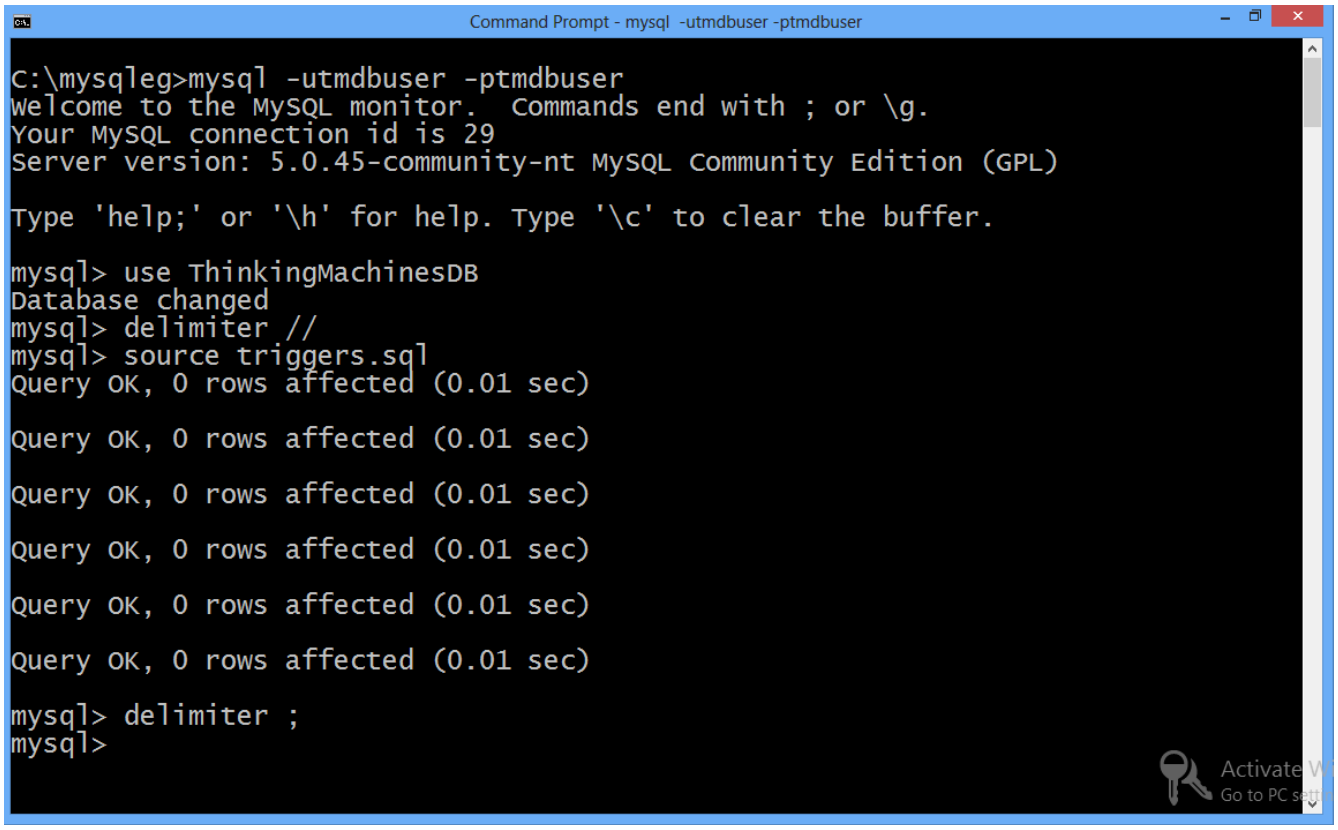
First of all quit from tmdbuser login, then login as root user using (mysql -uroot -pyoursurname)
grant super on *.*   to   'tmdbuser'@'%'
now quit from root user
Create a file named as triggers.sql with following sql statements.

triggers.sql

```
create trigger sale_item_insert after insert on sale_item for each row
begin
update item set total_sales=total_sales+new.quantity,closing_stock=closing_stock-new.quantity where
code=new.item_code;
end;//
create trigger sale_item_delete after delete on sale_item for each row
begin
update item set total_sales=total_sales-old.quantity,closing_stock=closing_stock+old.quantity where
code=old.item_code;
end;//
create trigger sale_item_update after update on sale_item for each row
begin
update item set total_sales=total_sales-old.quantity,closing_stock=closing_stock+old.quantity where
code=old.item_code;
update item set total_sales=total_sales+new.quantity,closing_stock=closing_stock-new.quantity where
code=new.item_code;
end;//
create trigger purchase_item_insert after insert on purchase_item for each row
begin
update item set
total_purchases=total_purchases+new.quantity,closing_stock=closing_stock+new.quantity where
code=new.item_code;
end;//
create trigger purchase_item_delete after delete on purchase_item for each row
begin
update item set total_purchases=total_purchases-old.quantity,closing_stock=closing_stock-old.quantity
where code=old.item_code;
end;//
create trigger purchase_item_update after update on purchase_item for each row
begin
update item set total_purchases=total_purchases-old.quantity,closing_stock=closing_stock-old.quantity
where code=old.item_code;
update item set
total_purchases=total_purchases+new.quantity,closing_stock=closing_stock+new.quantity where
code=new.item_code;
```

```
end;//
```

Now login into mysql as tmdbuser and type the following
delimiter //
then
source triggers.sql
if no errors, then
delimiter ;
Refer the following UI



```
C:\mysqleg>mysql -utmdbuser -ptmdbuser
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 5.0.45-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use ThinkingMachinesDB
Database changed
mysql> delimiter //
mysql> source triggers.sql
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql>
```

To get list of triggers type
show triggers\G

Now let us create sql files to insert sample data

item_data.sql

insert into item (name,opening_stock,closing_stock) values('Screw',1000,1000);
insert into item (name,opening_stock,closing_stock) values('Computer',1500,1500);
insert into item (name) values ('Printer');
insert into item (name) values ('Mouse pad');
insert into item (name) values ('Pencil');
insert into item (name,opening_stock,closing_stock) values ('Pencil Box',1800,1800);

Login into mysql using (tmdbuser) and type (source item_data.sql) to insert records

---

Now view the records from item table using (select * from item), note down the codes of the inserted records, in my case it is (1 to 6)

customer_data.sql

```
insert into customer (name) values ('Sameer');
insert into customer (name) values ('Rakesh');
insert into customer (name) values ('Mohan');
```

Login into mysql using (tmdbuser) and type (source customer_data.sql) to insert records

Now view the records from item table using (select * from customer), note down the codes of the inserted records, in my case it is (1 to 3)

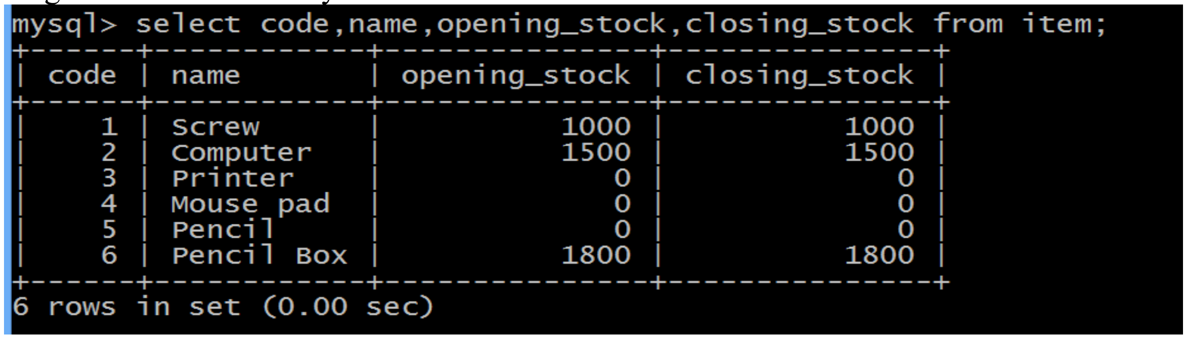
supplier_data.sql

```
insert into supplier (name) values ('Sam');
insert into supplier (name) values ('Tom');
insert into supplier (name) values ('John');
insert into supplier (name) values ('Joy');
insert into supplier (name) values ('Tony');
```

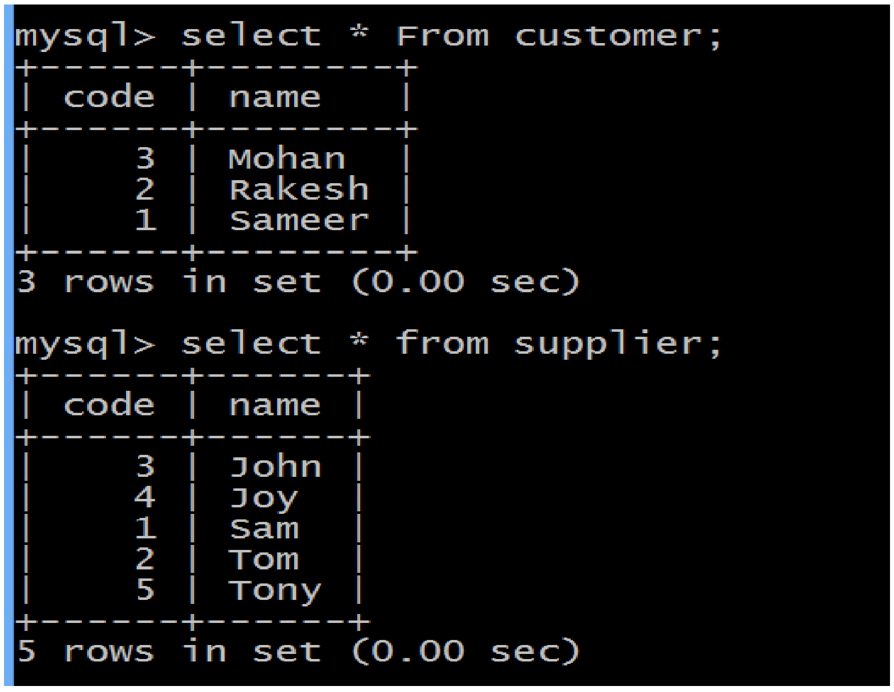
Login into mysql using (tmdbuser) and type (source supplier_data.sql) to insert records

Now view the records from item table using (select * from supplier), note down the codes of the inserted records, in my case it is (1 to 5)

Following are the records at my end



```
mysql> select code,name,opening_stock,closing_stock from item;
+------+------------+---------------+---------------+
| code | name       | opening_stock | closing_stock |
+------+------------+---------------+---------------+
|    1 | Screw      |          1000 |          1000 |
|    2 | Computer   |          1500 |          1500 |
|    3 | Printer    |             0 |             0 |
|    4 | Mouse pad  |             0 |             0 |
|    5 | Pencil     |             0 |             0 |
|    6 | Pencil Box |          1800 |          1800 |
+------+------------+---------------+---------------+
6 rows in set (0.00 sec)
```



```
mysql> select * From customer;
+------+--------+
| code | name   |
+------+--------+
|    3 | Mohan  |
|    2 | Rakesh |
|    1 | Sameer |
+------+--------+
3 rows in set (0.00 sec)

mysql> select * from supplier;
+------+------+
| code | name |
+------+------+
|    3 | John |
|    4 | Joy  |
|    1 | Sam  |
|    2 | Tom  |
|    5 | Tony |
+------+------+
5 rows in set (0.00 sec)
```

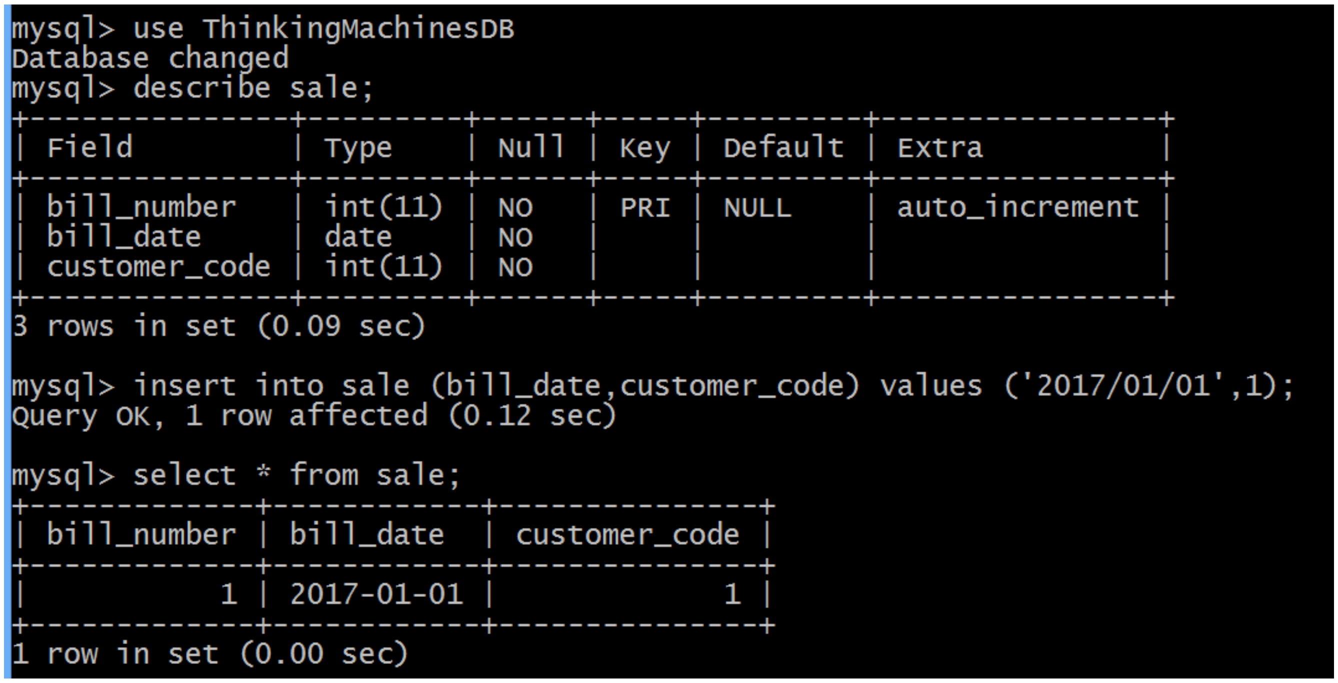
Now you need to insert record in sale table, get the bill_number assigned to the inserted record and then insert some records in sale_item table and then see the effect of trigger on item table.

Then update some records of the sale_item table and again see the effect of trigger on item table

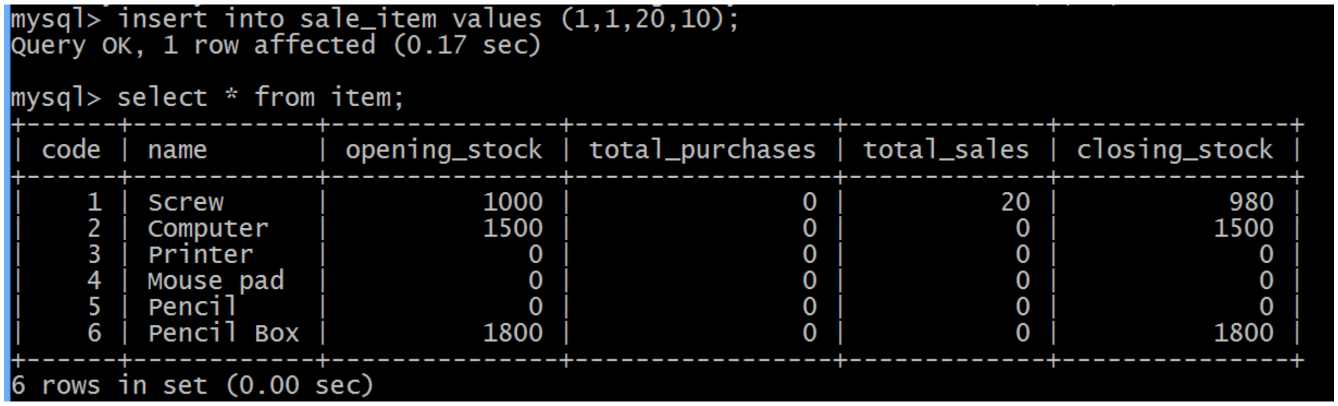
Then delete some records of the sale_item table and again see the effect of trigger on item table

The do the same for purchase and purchase_item table.

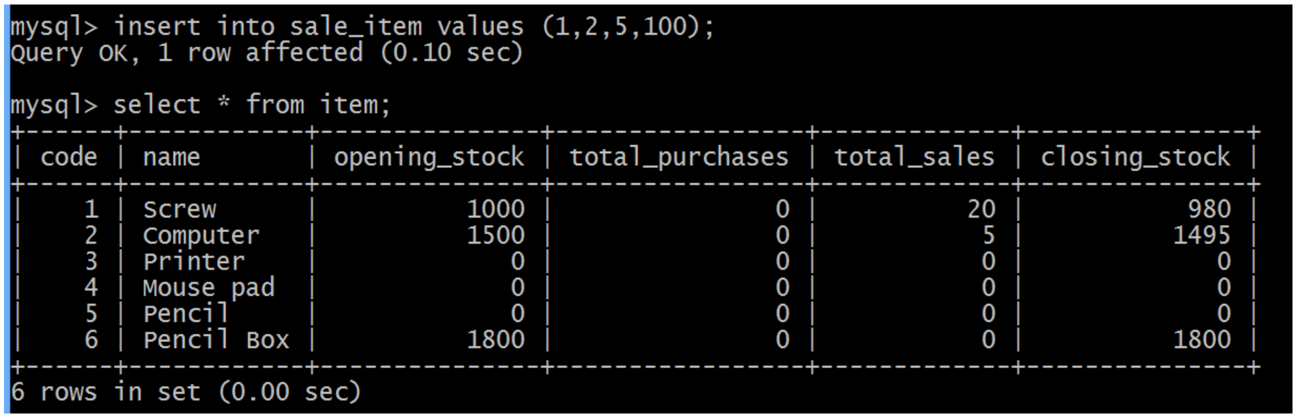
I am attaching screen shots from my end.



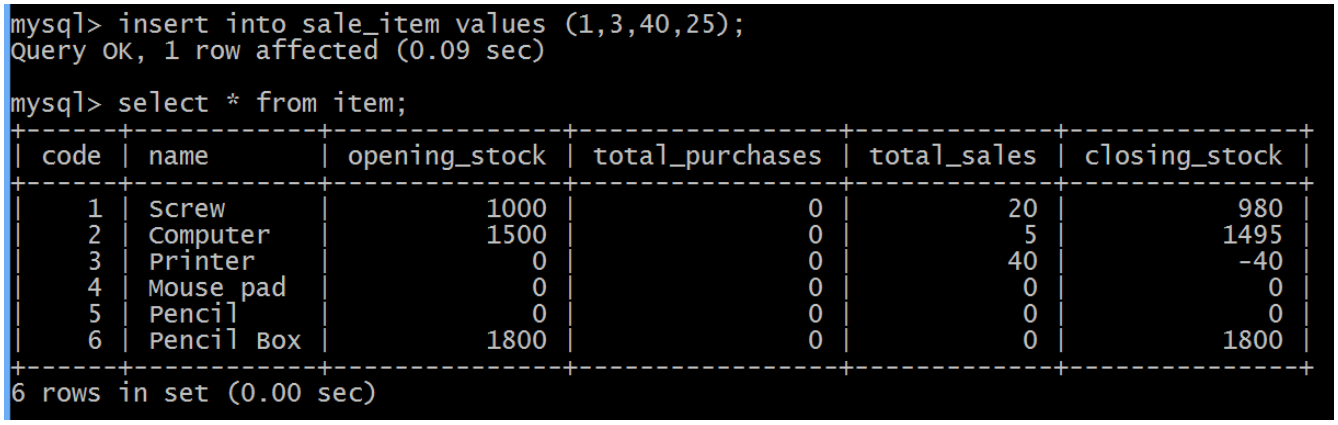
```
mysql> use ThinkingMachinesDB
Database changed
mysql> describe sale;
+---------------+---------+------+-----+---------+----------------+
| Field         | Type    | Null | Key | Default | Extra          |
+---------------+---------+------+-----+---------+----------------+
| bill_number   | int(11) | NO   | PRI | NULL    | auto_increment |
| bill_date     | date    | NO   |     |         |                |
| customer_code | int(11) | NO   |     |         |                |
+---------------+---------+------+-----+---------+----------------+
3 rows in set (0.09 sec)

mysql> insert into sale (bill_date,customer_code) values ('2017/01/01',1);
Query OK, 1 row affected (0.12 sec)

mysql> select * from sale;
+-------------+------------+---------------+
| bill_number | bill_date  | customer_code |
+-------------+------------+---------------+
|           1 | 2017-01-01 |             1 |
+-------------+------------+---------------+
1 row in set (0.00 sec)
```



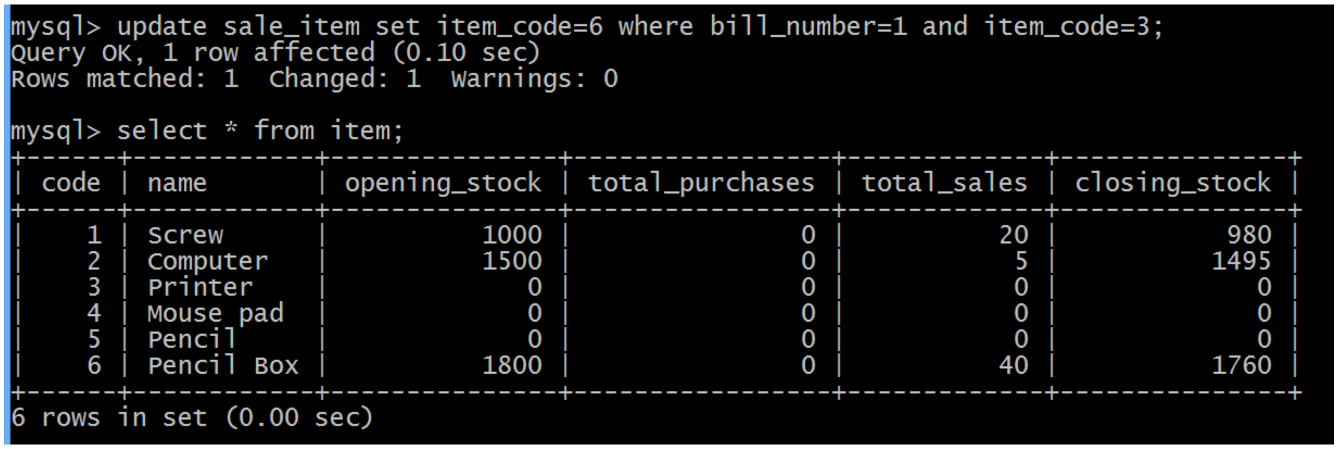
```
mysql> insert into sale_item values (1,1,20,10);
Query OK, 1 row affected (0.17 sec)

mysql> select * from item;
+------+------------+---------------+-----------------+-------------+---------------+
| code | name       | opening_stock | total_purchases | total_sales | closing_stock |
+------+------------+---------------+-----------------+-------------+---------------+
|    1 | Screw      |          1000 |               0 |          20 |           980 |
|    2 | Computer   |          1500 |               0 |           0 |          1500 |
|    3 | Printer    |             0 |               0 |           0 |             0 |
|    4 | Mouse pad  |             0 |               0 |           0 |             0 |
|    5 | Pencil     |             0 |               0 |           0 |             0 |
|    6 | Pencil Box |          1800 |               0 |           0 |          1800 |
+------+------------+---------------+-----------------+-------------+---------------+
6 rows in set (0.00 sec)
```



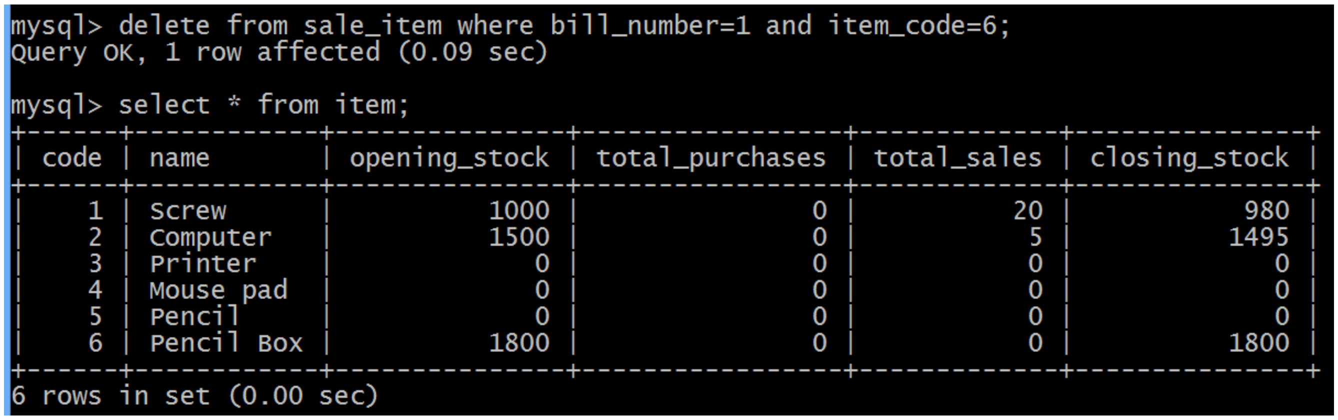
```
mysql> insert into sale_item values (1,2,5,100);
Query OK, 1 row affected (0.10 sec)

mysql> select * from item;
+------+------------+---------------+-----------------+-------------+---------------+
| code | name       | opening_stock | total_purchases | total_sales | closing_stock |
+------+------------+---------------+-----------------+-------------+---------------+
|    1 | Screw      |          1000 |               0 |          20 |           980 |
|    2 | Computer   |          1500 |               0 |           5 |          1495 |
|    3 | Printer    |             0 |               0 |           0 |             0 |
|    4 | Mouse pad  |             0 |               0 |           0 |             0 |
|    5 | Pencil     |             0 |               0 |           0 |             0 |
|    6 | Pencil Box |          1800 |               0 |           0 |          1800 |
+------+------------+---------------+-----------------+-------------+---------------+
6 rows in set (0.00 sec)
```



```
mysql> insert into sale_item values (1,3,40,25);
Query OK, 1 row affected (0.09 sec)

mysql> select * from item;
+------+------------+---------------+-----------------+-------------+---------------+
| code | name       | opening_stock | total_purchases | total_sales | closing_stock |
+------+------------+---------------+-----------------+-------------+---------------+
|    1 | Screw      |          1000 |               0 |          20 |           980 |
|    2 | Computer   |          1500 |               0 |           5 |          1495 |
|    3 | Printer    |             0 |               0 |          40 |           -40 |
|    4 | Mouse pad  |             0 |               0 |           0 |             0 |
|    5 | Pencil     |             0 |               0 |           0 |             0 |
|    6 | Pencil Box |          1800 |               0 |           0 |          1800 |
+------+------------+---------------+-----------------+-------------+---------------+
6 rows in set (0.00 sec)
```



```
mysql> update sale_item set item_code=6 where bill_number=1 and item_code=3;
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from item;
+------+------------+---------------+-----------------+-------------+---------------+
| code | name       | opening_stock | total_purchases | total_sales | closing_stock |
+------+------------+---------------+-----------------+-------------+---------------+
|    1 | Screw      |          1000 |               0 |          20 |           980 |
|    2 | Computer   |          1500 |               0 |           5 |          1495 |
|    3 | Printer    |             0 |               0 |           0 |             0 |
|    4 | Mouse pad  |             0 |               0 |           0 |             0 |
|    5 | Pencil     |             0 |               0 |           0 |             0 |
|    6 | Pencil Box |          1800 |               0 |          40 |          1760 |
+------+------------+---------------+-----------------+-------------+---------------+
6 rows in set (0.00 sec)
```

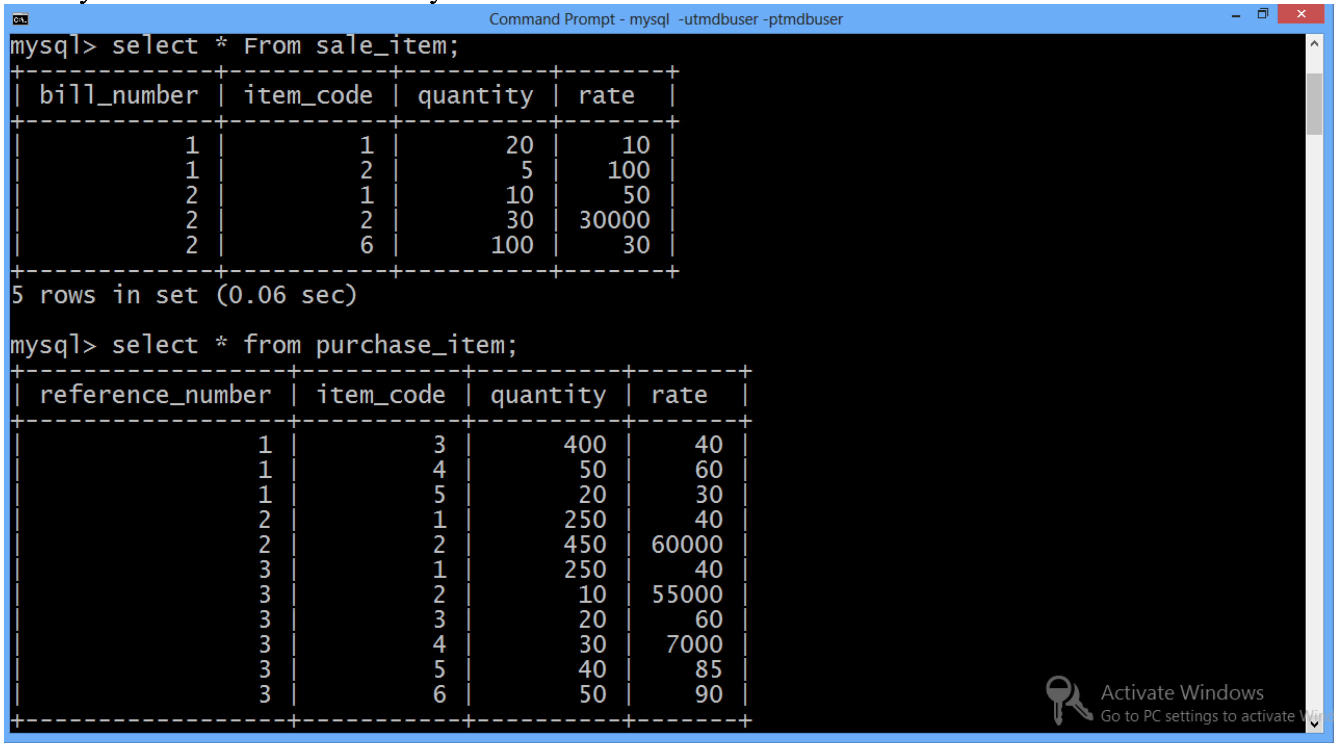


```
mysql> delete from sale_item where bill_number=1 and item_code=6;
Query OK, 1 row affected (0.09 sec)

mysql> select * from item;
+------+------------+---------------+-----------------+-------------+---------------+
| code | name       | opening_stock | total_purchases | total_sales | closing_stock |
+------+------------+---------------+-----------------+-------------+---------------+
|    1 | Screw      |          1000 |               0 |          20 |           980 |
|    2 | Computer   |          1500 |               0 |           5 |          1495 |
|    3 | Printer    |             0 |               0 |           0 |             0 |
|    4 | Mouse pad  |             0 |               0 |           0 |             0 |
|    5 | Pencil     |             0 |               0 |           0 |             0 |
|    6 | Pencil Box |          1800 |               0 |           0 |          1800 |
+------+------------+---------------+-----------------+-------------+---------------+
6 rows in set (0.00 sec)
```

Similarly perform more operations as per your requirement. I am pasting all the sql statements fired by me as follows.

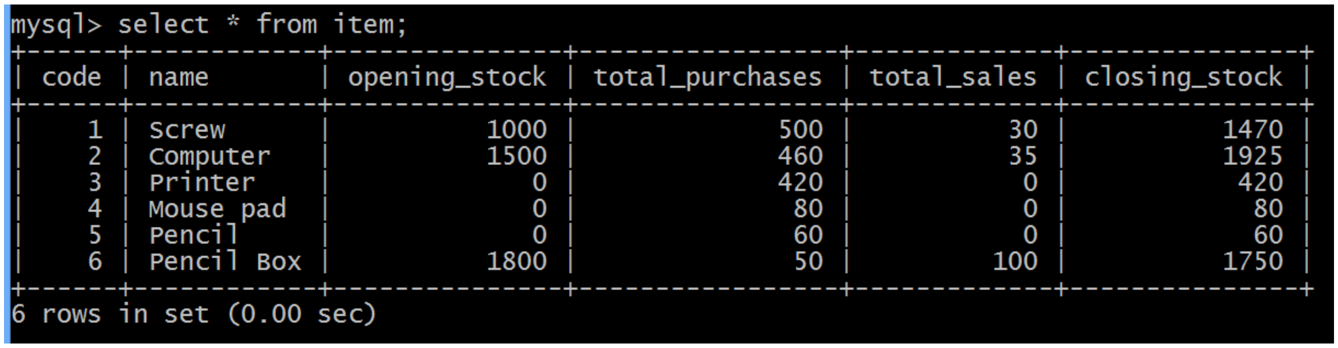
```
insert into sale (bill_date,customer_code) values ('2017/01/02',3);
insert into sale_item values(2,1,10,50);
insert into sale_item values(2,2,30,30000);
insert into sale_item values(2,6,100,30);
insert into purchase (bill_number,bill_date,supplier_code) values ('Jan 01/2017','2017/01/01',2);
insert into purchase_item values(1,3,400,40);
insert into purchase_item values(1,4,50,60);
insert into purchase_item values(1,5,20,30);
insert into purchase (bill_number,bill_date,supplier_code) values ('309','2017/01/01',4);
insert into purchase_item values(2,1,250,40);
insert into purchase_item values(2,2,450,60000);
insert into purchase (bill_number,bill_date,supplier_code) values ('504','2017/01/01',3);
insert into purchase_item values(3,1,250,40);
insert into purchase_item values(3,2,10,55000);
insert into purchase_item values(3,3,20,60);
insert into purchase_item values(3,4,30,7000);
insert into purchase_item values(3,5,40,85);
insert into purchase_item values(3,6,50,90);
```

Finally these are the records at my end



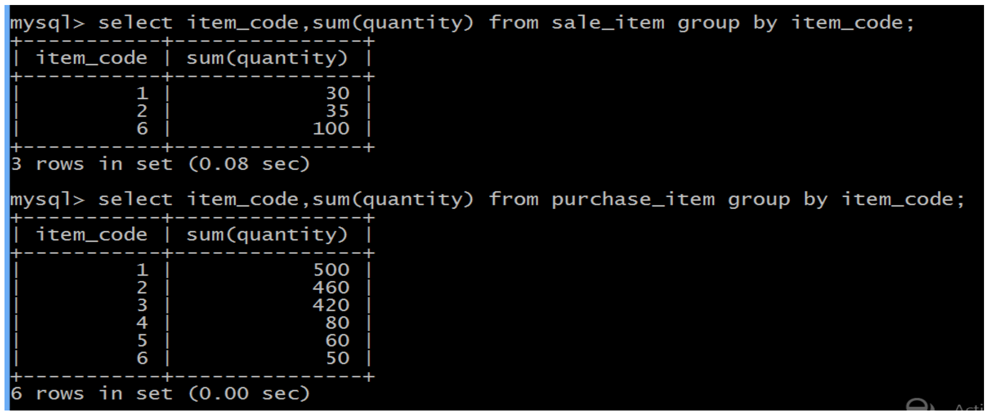
Command Prompt - mysql -utmdbuser -ptmdbuser

```
mysql> select * From sale_item;
+-------------+-----------+----------+-------+
| bill_number | item_code | quantity | rate  |
+-------------+-----------+----------+-------+
|           1 |         1 |       20 |    10 |
|           1 |         2 |        5 |   100 |
|           2 |         1 |       10 |    50 |
|           2 |         2 |       30 | 30000 |
|           2 |         6 |      100 |    30 |
+-------------+-----------+----------+-------+
5 rows in set (0.06 sec)

mysql> select * from purchase_item;
+------------------+-----------+----------+-------+
| reference_number | item_code | quantity | rate  |
+------------------+-----------+----------+-------+
|                1 |         3 |      400 |    40 |
|                1 |         4 |       50 |    60 |
|                1 |         5 |       20 |    30 |
|                2 |         1 |      250 |    40 |
|                2 |         2 |      450 | 60000 |
|                3 |         1 |      250 |    40 |
|                3 |         2 |       10 | 55000 |
|                3 |         3 |       20 |    60 |
|                3 |         4 |       30 |  7000 |
|                3 |         5 |       40 |    85 |
|                3 |         6 |       50 |    90 |
+------------------+-----------+----------+-------+
```



```
mysql> select * from item;
+------+------------+---------------+-----------------+-------------+---------------+
| code | name       | opening_stock | total_purchases | total_sales | closing_stock |
+------+------------+---------------+-----------------+-------------+---------------+
|    1 | Screw      |          1000 |             500 |          30 |          1470 |
|    2 | Computer   |          1500 |             460 |          35 |          1925 |
|    3 | Printer    |             0 |             420 |           0 |           420 |
|    4 | Mouse pad  |             0 |              80 |           0 |            80 |
|    5 | Pencil     |             0 |              60 |           0 |            60 |
|    6 | Pencil Box |          1800 |              50 |         100 |          1750 |
+------+------------+---------------+-----------------+-------------+---------------+
6 rows in set (0.00 sec)
```

Now lets verify that the calculations are correct



```
mysql> select item_code,sum(quantity) from sale_item group by item_code;
+-----------+---------------+
| item_code | sum(quantity) |
+-----------+---------------+
|         1 |            30 |
|         2 |            35 |
|         6 |           100 |
+-----------+---------------+
3 rows in set (0.08 sec)

mysql> select item_code,sum(quantity) from purchase_item group by item_code;
+-----------+---------------+
| item_code | sum(quantity) |
+-----------+---------------+
|         1 |           500 |
|         2 |           460 |
|         3 |           420 |
|         4 |            80 |
|         5 |            60 |
|         6 |            50 |
+-----------+---------------+
6 rows in set (0.00 sec)
```

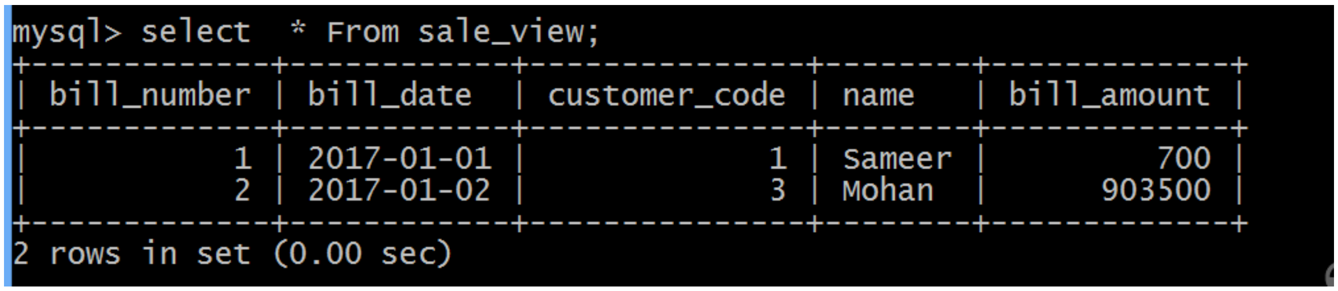
Now you do some calculations to check of the records of item table has correct information. Everything

is correct at my end.
Now let us create some views.
Note : I will be creating separate sql files, I assume that you already know how to run its contents.

sale_view.sql

```
create view sale_view
as
select sale.bill_number,sale.bill_date,sale.customer_code,customer.name,sum(quantity*rate) as
bill_amount from
sale,customer,sale_item where sale.customer_code=customer.code and
sale.bill_number=sale_item.bill_number
group by sale.bill_number,sale.bill_date,sale.customer_code;
```

After creating the view you can type
select * from sale_view

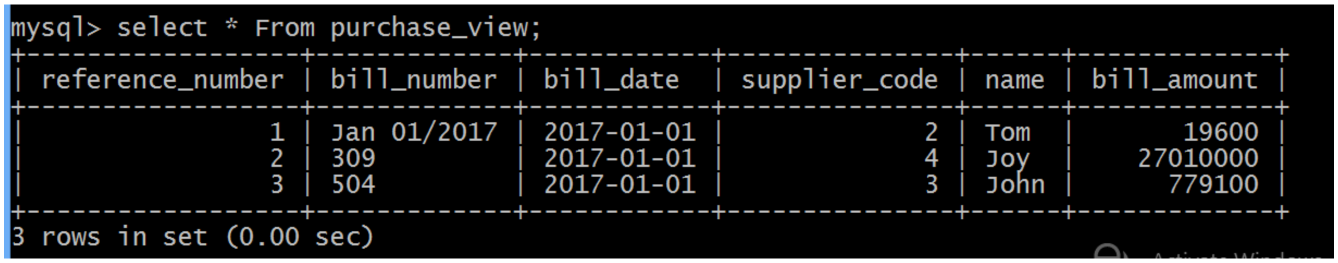


```
mysql> select  * From sale_view;
+-------------+------------+---------------+--------+-------------+
| bill_number | bill_date  | customer_code | name   | bill_amount |
+-------------+------------+---------------+--------+-------------+
|           1 | 2017-01-01 |             1 | Sameer |         700 |
|           2 | 2017-01-02 |             3 | Mohan  |      903500 |
+-------------+------------+---------------+--------+-------------+
2 rows in set (0.00 sec)
```

purchase_view.sql

```
create view purchase_view
as
select
purchase.reference_number,purchase.bill_number,purchase.bill_date,purchase.supplier_code,supplier.n
ame,sum(quantity*rate) as bill_amount from
purchase,supplier,purchase_item where purchase.supplier_code=supplier.code and
purchase.reference_number=purchase_item.reference_number
group by purchase.reference_number,purchase.bill_number,purchase.bill_date,purchase.supplier_code;
```

After creating the view you can type
select *   from purchase_view

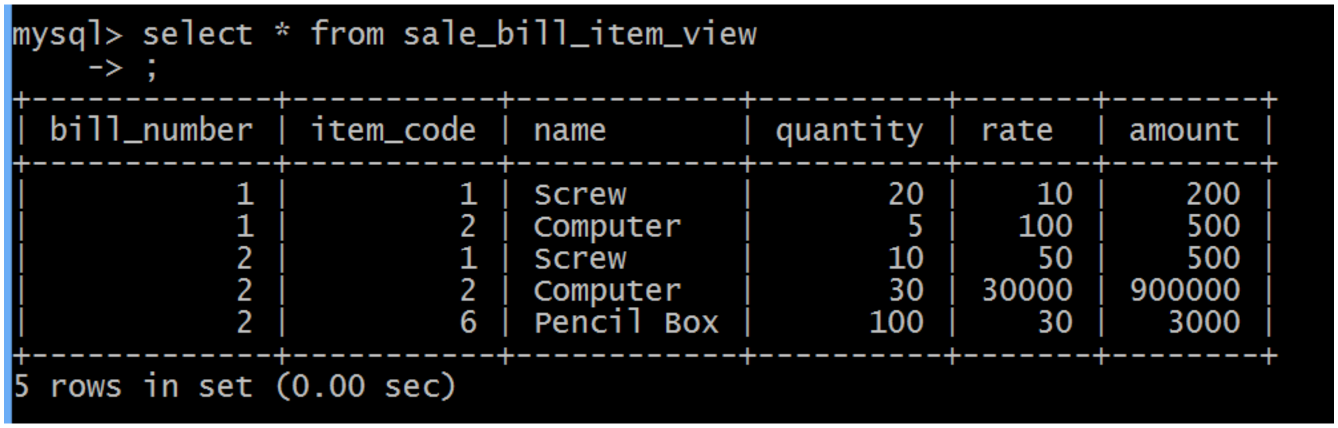


```
mysql> select * From purchase_view;
+------------------+-------------+------------+---------------+------+-------------+
| reference_number | bill_number | bill_date  | supplier_code | name | bill_amount |
+------------------+-------------+------------+---------------+------+-------------+
|                1 | Jan 01/2017 | 2017-01-01 |             2 | Tom  |       19600 |
|                2 | 309         | 2017-01-01 |             4 | Joy  |    27010000 |
|                3 | 504         | 2017-01-01 |             3 | John |      779100 |
+------------------+-------------+------------+---------------+------+-------------+
3 rows in set (0.00 sec)
```

sale_bill_item_view.sql

```
create view sale_bill_item_view
as
select
sale_item.bill_number,sale_item.item_code,item.name,sale_item.quantity,sale_item.rate,sale_item.quantity*sale_item.rate as amount from
sale_item inner join item on sale_item.item_code=item.code order by sale_item.bill_number;
```

After creating the view you can type
select * from sale_bill_item_view

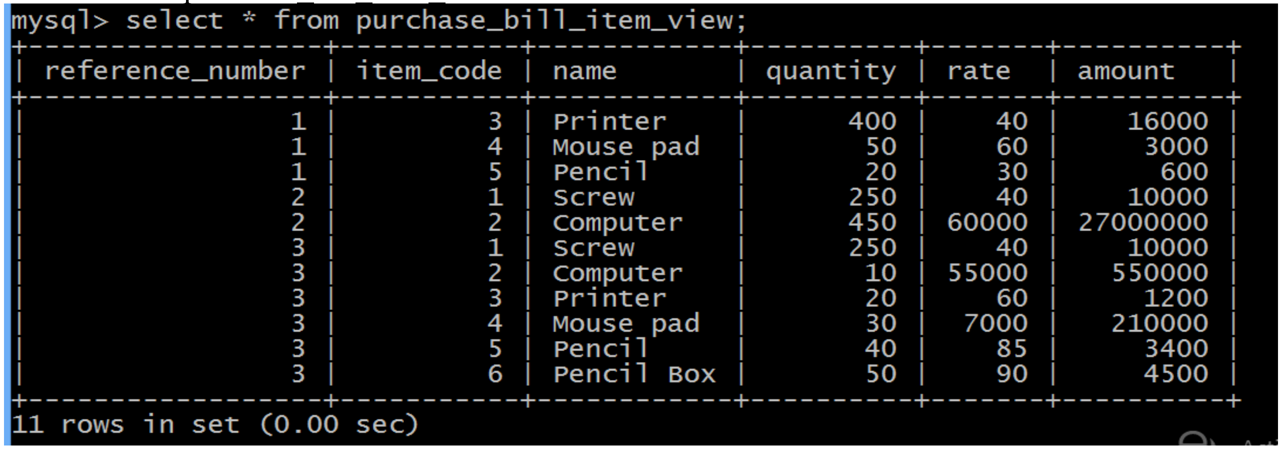


```
mysql> select * from sale_bill_item_view
    -> ;
+-------------+-----------+------------+----------+-------+--------+
| bill_number | item_code | name       | quantity | rate  | amount |
+-------------+-----------+------------+----------+-------+--------+
|           1 |         1 | Screw      |       20 |    10 |    200 |
|           1 |         2 | Computer   |        5 |   100 |    500 |
|           2 |         1 | Screw      |       10 |    50 |    500 |
|           2 |         2 | Computer   |       30 | 30000 | 900000 |
|           2 |         6 | Pencil Box |      100 |    30 |   3000 |
+-------------+-----------+------------+----------+-------+--------+
5 rows in set (0.00 sec)
```

purchase_bill_item_view.sql

```
create view purchase_bill_item_view
as
select
purchase_item.reference_number,purchase_item.item_code,item.name,purchase_item.quantity,purchase_item.rate,purchase_item.quantity*purchase_item.rate as amount from
purchase_item inner join item on purchase_item.item_code=item.code order by
purchase_item.reference_number;
```

After creating the view you can type
select * from purchase_bill_item_view



```
mysql> select * from purchase_bill_item_view;
+------------------+-----------+------------+----------+-------+----------+
| reference_number | item_code | name       | quantity | rate  | amount   |
+------------------+-----------+------------+----------+-------+----------+
|                1 |         3 | Printer    |      400 |    40 |    16000 |
|                1 |         4 | Mouse pad  |       50 |    60 |     3000 |
|                1 |         5 | Pencil     |       20 |    30 |      600 |
|                2 |         1 | Screw      |      250 |    40 |    10000 |
|                2 |         2 | Computer   |      450 | 60000 | 27000000 |
|                3 |         1 | Screw      |      250 |    40 |    10000 |
|                3 |         2 | Computer   |       10 | 55000 |   550000 |
|                3 |         3 | Printer    |       20 |    60 |     1200 |
|                3 |         4 | Mouse pad  |       30 |  7000 |   210000 |
|                3 |         5 | Pencil     |       40 |    85 |     3400 |
|                3 |         6 | Pencil Box |       50 |    90 |     4500 |
+------------------+-----------+------------+----------+-------+----------+
11 rows in set (0.00 sec)
```

Now let us write a java code to print all information about sale bills.

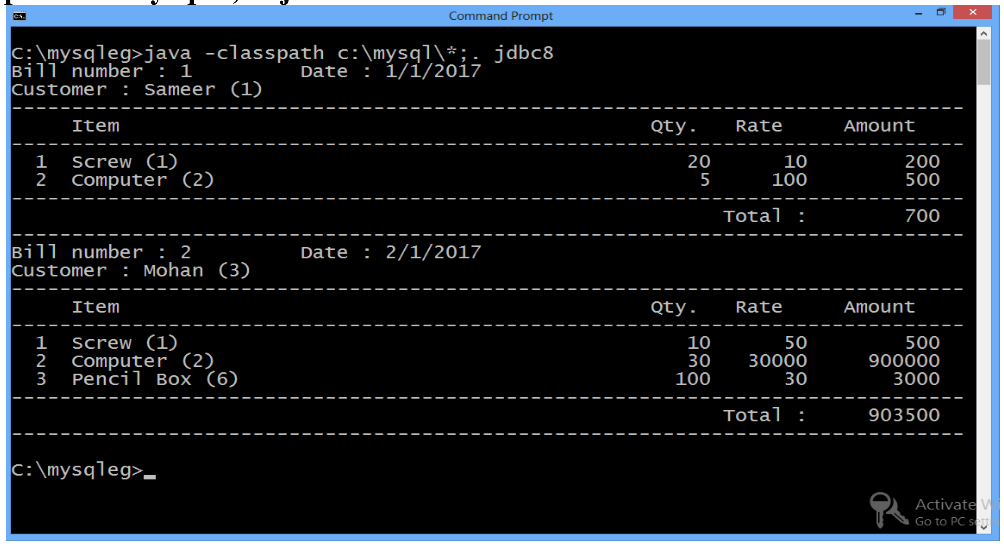
First of all create a folder named as mysql on c:\
download mysql.jar and save it to c:\mysql

jdbc8.java (will compile)

```
import java.sql.*;
class jdbc8
{
public static void main(String kk[])
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection c;
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
Statement s=c.createStatement();
int billNumber;
Date billDate;
int customerCode;
String customerName;
int billAmount;
int itemCode;
String itemName;
int quantity;
int rate;
int amount;
int sno;
PreparedStatement ps;
ResultSet r2;
ResultSet r1;
r1=s.executeQuery("select * from sale_view order by bill_number");
while(r1.next())
{
billNumber=r1.getInt("bill_number");
billDate=r1.getDate("bill_date");
customerCode=r1.getInt("customer_code");
customerName=r1.getString("name").trim();
billAmount=r1.getInt("bill_amount");
System.out.println("Bill number : "+billNumber+"\t\tDate : "+(billDate.getDate()+"/"+
(billDate.getMonth()+1)+"/"+(billDate.getYear()+1900)));
System.out.printf("Customer : %s (%d)\n",customerName,customerCode);
System.out.println("-----------------------------------------------------------------------------");
System.out.println("    Item                                    Qty.   Rate    Amount");
System.out.println("-----------------------------------------------------------------------------");
ps=c.prepareStatement("select * from sale_bill_item_view where bill_number=?");
ps.setInt(1,billNumber);
r2=ps.executeQuery();
sno=0;
while(r2.next())
{
sno++;
itemCode=r2.getInt("item_code");
itemName=r2.getString("name").trim();
quantity=r2.getInt("quantity");
rate=r2.getInt("rate");
amount=r2.getInt("amount");
System.out.printf("%3d  %-45s %7d %7d %10d\n",sno,itemName+"
("+itemCode+")",quantity,rate,amount);
}
r2.close();
ps.close();
System.out.println("--------------------------------------------------------------------------------");
System.out.printf("%64s : %10d\n","Total",billAmount);
System.out.println("--------------------------------------------------------------------------------");
}
r1.close();
s.close();
c.close();
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

to run the above code, type

java -classpath c:\mysql\*;. jdbc8



```
C:\mysqleg>java -classpath c:\mysql\*;. jdbc8
Bill number : 1         Date : 1/1/2017
Customer : Sameer (1)
--------------------------------------------------------------------------------
     Item                                             Qty.   Rate      Amount
--------------------------------------------------------------------------------
  1  Screw (1)                                          20      10         200
  2  Computer (2)                                        5     100         500
--------------------------------------------------------------------------------
                                                            Total :        700
--------------------------------------------------------------------------------
Bill number : 2         Date : 2/1/2017
Customer : Mohan (3)
--------------------------------------------------------------------------------
     Item                                             Qty.   Rate      Amount
--------------------------------------------------------------------------------
  1  Screw (1)                                          10      50         500
  2  Computer (2)                                       30   30000      900000
  3  Pencil Box (6)                                    100      30        3000
--------------------------------------------------------------------------------
                                                            Total :     903500
--------------------------------------------------------------------------------

C:\mysqleg>
```

Do something similar to print all purchase bill data with reference number.

MySQL – Creating procedures/functions

create a file named as add_customer.sql

add_customer.sql

```
create function add_customer(name char(50)) returns int
Begin
declare is_error int default 0;
DECLARE CONTINUE HANDLER FOR SQLEXCEPTION set is_error=-1;
insert into customer(name) values(name);
if is_error=0 then
return LAST_INSERT_ID();
else
return -1;
end if;
end; //
```

To create the function, login into mysql (tmdbuser) account
first of all type
delimiter //
then type
source add_customer.sql
then type
delimiter ;

Now let us create java code to call the function

jdbc9.java (will compile)

```
import java.sql.*;
class jdbc9
{
public static void main(String data[])
{
try
{
String name=data[0];
Class.forName("com.mysql.jdbc.Driver");
Connection c;
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
CallableStatement cs=c.prepareCall("{? = call add_customer(?)}");
cs.registerOutParameter(1,Types.INTEGER);
cs.setString(2,name);
cs.execute();
Integer customerCode=cs.getInt(1);
c.close();
if(customerCode==-1)
{
System.out.println("Customer not inserted");
}
else
{
System.out.println("Customer inserted successfully with code as : "+customerCode);
}

}catch(Exception e)
{
System.out.println(e);
}
}
}
```

---

compile the above code and to run type

```
java   -classpath   c:\mysql\*;.   Jdbc9   someName
```

Note : replace someName with name of your choice, try adding duplicate name and see what happens

---

Now let us create function to update customer

update_customer.sql

```
create function update_customer(oldName char(50),newName char(50)) returns boolean
Begin
declare updated Boolean default true;
declare v_code int;
DECLARE CONTINUE HANDLER FOR SQLEXCEPTION set updated=false;
select code into v_code from customer where name=oldName limit 1;
if v_code is NULL then
set updated=false;
else
update customer set name=newName where code=v_code;
end if;
return updated;
end; //
```

To run, first login into mysql (tmdbuser)
then type
delimiter //
then type
source update_customer.sql
then type
delimiter ;

---

java code to call update_customer function

jdbc10.java (will compile)

```
import java.sql.*;
class jdbc10
{
public static void main(String data[])
{
try
{
String oldName=data[0];
String newName=data[1];
Class.forName("com.mysql.jdbc.Driver");
Connection c;
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
CallableStatement cs=c.prepareCall("{? = call update_customer(?,?)}");
cs.registerOutParameter(1,Types.BOOLEAN);
cs.setString(2,oldName);
cs.setString(3,newName);
cs.execute();
Boolean updated=cs.getBoolean(1);
c.close();
if(!updated)
{
System.out.println("Customer not updated");
}
else
{
System.out.println("Customer updated");
}
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

Compile & Run the above code. (Pass 2 arguments – oldName and newName). You can then drop function update_customer and then recreate it to accept code and newName instead of oldName and newName and then modify the jdbc code to pass customerCode and newName as arguments.

Similarly create (delete_customer.sql with delete_customer function) and jdbc code to call the delete_customer function. The function should accept customerCode as argument.

Now let us create a procedure
Create a file named as add_supplier.sql

add_supplier.sql

```
create procedure add_supplier(v_name char(50))
Begin
declare cnt int default 0;
select count(*) into cnt from supplier where name=v_name;
if cnt>0 then
signal SQLSTATE '45000' set MESSAGE_TEXT = 'Supplier exists';
else
insert into supplier (name) values(v_name);
end if;
end; //
```

login into mysql (tmdbuser)
then type

delimiter //
then type
source add_supplier.sql
then type
delimiter ;
then type
call add_supplier('Varun');
then type
call add_supplier('Shankar');
then type
call add_supplier('Varun');
you should see the error message (Supplier Exists)

Now the jdbc code to call the procedure.

jdbc11.java (will compile)

```
import java.sql.*;
class jdbc11
{
public static void main(String data[])
{
Connection c=null;
try
{
String name=data[0];
Class.forName("com.mysql.jdbc.Driver");
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
CallableStatement cs=c.prepareCall("{call add_supplier(?)}");
cs.setString(1,name);
cs.execute();
System.out.println("Supplier added");
}catch(Exception e)
{
System.out.println(e);
}
finally
{
try
{
c.close();
}catch(Exception m)
{
System.out.println(m);
}
}
}
}
```

Compile the above code and run as done earlier. Add some supplier and try adding duplicate suppliers and you should see the message, SQLException : Supplier Exists

---

creating scrollable and updatable ResultSet

jdbc12.java (will compile)

```
import java.sql.*;
class jdbc12
{
public static void main(String kk[])
{
Connection c=null;
try
{
Class.forName("com.mysql.jdbc.Driver");
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
Statement s=c.createStatement();
ResultSet r=s.executeQuery("select * from item order by name");
int code;
String name;
boolean b;
b=r.next();
System.out.println(b);
if(b)
{
code=r.getInt("code");
name=r.getString("name").trim();
System.out.printf("Code %d, Name %s\n",code,name);
}
b=r.next();
System.out.println(b);
if(b)
{
code=r.getInt("code");
name=r.getString("name").trim();
System.out.printf("Code %d, Name %s\n",code,name);
}
b=r.previous();
System.out.println(b);
if(b)
{
code=r.getInt("code");
name=r.getString("name").trim();
System.out.printf("Code %d, Name %s\n",code,name);
}
b=r.previous();
System.out.println(b);
if(b)
{
code=r.getInt("code");
name=r.getString("name").trim();
System.out.printf("Code %d, Name %s\n",code,name);
}

r.close();
s.close();
}catch(Exception e)
{
System.out.println(e);
}
finally
{
try
{
c.close();
}catch(Exception m)
{
System.out.println(m);
}
}
}
}
```

jdbc13.java (will compile)

```
import java.sql.*;
import java.io.*;
class jdbc13
{
public static void main(String gg[])
{
Connection c=null;
try
{
Class.forName("com.mysql.jdbc.Driver");
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
Statement s=c.createStatement();
s.executeUpdate("insert into item (name) values('Laptop')");
s.executeUpdate("insert into item (name) values('Pen Drive')");
s.executeUpdate("insert into item (name) values('Sharpner')");
s.executeUpdate("insert into item (name) values('Eraser')");
s.executeUpdate("insert into item (name) values('Pen')");
s.executeUpdate("insert into item (name) values('Ink Bottle')");
s.executeUpdate("insert into item (name) values('Scale')");
```

```
s.executeUpdate("insert into item (name) values('Slider')");
s.executeUpdate("insert into item (name) values('Duster')");
s.executeUpdate("insert into item (name) values('Marker')");
s.close();
s=c.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);
ResultSet r=s.executeQuery("select * from item");
r.absolute(4);
int rowNum = r.getRow();
System.out.println("rowNum should be 4 " + rowNum);
r.relative(-2);
rowNum = r.getRow();
System.out.println("rowNum should be 2 " + rowNum);
r.relative(1);
rowNum = r.getRow();
System.out.println("rowNum should be 3 " + rowNum);
r.last();
System.out.println("after last? " + r.isAfterLast() );
r.next();
System.out.println("after last? " + r.isAfterLast() );
int code;
String name;
if (!r.isAfterLast())
{
code = r.getInt("code");
name = r.getString("name").trim();
System.out.printf("Code : %d, Name %s\n",code,name);
}
System.out.println("------------------------------------------------");
r.afterLast();
while (r.previous())
{
code = r.getInt("code");
name = r.getString("name").trim();
System.out.printf("Code : %d, Name %s\n",code,name);
}
r.close();
s.close();
System.out.println("------------------------------------------------");
s=c.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
r=s.executeQuery("SELECT * from item");
while(r.next())
{
code=r.getInt("code");
name=r.getString("name").trim();
if(name.equals("Slider"))
{
r.updateString("name","Set square");
r.updateRow();
}
}
r.close();
s.close();
System.out.println("After Updation");
s=c.createStatement();
r=s.executeQuery("select * from item");
while(r.next())
{
code=r.getInt("code");
name = r.getString("name").trim();
System.out.printf("Code : %d, Name %s\n",code,name);
}
r.close();
s.close();
}catch(Exception e)
{
System.out.println(e);
}
finally
{
try
{
c.close();
}catch(Exception m)
{
System.out.println(m);
}
}
}
}
```

Backing up MySQL Database
jdbc14.java (will compile)

```
import java.sql.*;
class jdbc14
{
public static void main(String kk[])
{
String baseDirectory=null;
Connection c=null;
try
{
Class.forName("com.mysql.jdbc.Driver");
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
Statement s=c.createStatement();
ResultSet r=s.executeQuery("select @@BASEDIR as base_directory");
if(r.next())
{
baseDirectory=r.getString("base_directory").trim();
}
r.close();
s.close();
}catch(Exception e)
{
System.out.println(e);
}
finally
{
try
{
c.close();
}catch(Exception m)
{
System.out.println(m);
}
if(baseDirectory==null)
{
System.out.println("Cannot extract information about MySql");
}
else
{
System.out.println(baseDirectory);
}
}
}
}
```

jdbc15.java (will compile)

```
import java.sql.*;
import java.io.*;
class jdbc15
{
public static void main(String data[])
{
String backupFileName=data[0];
String baseDirectory=null;
Connection c=null;
try
{
Class.forName("com.mysql.jdbc.Driver");
c=DriverManager.getConnection("jdbc:mysql://localhost:3306/ThinkingMachinesDB","tmdbuser","tm
dbuser");
Statement s=c.createStatement();
ResultSet r=s.executeQuery("select @@BASEDIR as base_directory");
if(r.next())
{
baseDirectory=r.getString("base_directory").trim();
}
r.close();
s.close();
}catch(Exception e)
{
System.out.println(e);
}
finally
{
try
{
c.close();
}catch(Exception m)
{
System.out.println(m);
}
if(baseDirectory==null)
{
System.out.println("Cannot extract information about MySql, hence cannot take backup.");
return;
}
baseDirectory=baseDirectory;
File outputFile=new File(backupFileName);
if(outputFile.exists()) outputFile.delete();
File errorFile=new File("errors.err");
if(errorFile.exists()) errorFile.delete();
String command=baseDirectory+"bin/mysqldump.exe";
ProcessBuilder processBuilder=new ProcessBuilder(command,"-uroot","-
pkelkar","ThinkingMachinesDB");
processBuilder.redirectOutput(outputFile);
processBuilder.redirectError(errorFile);
try
{
Process process=processBuilder.start();
if(outputFile.exists()==false)
{
System.out.println("Unable to take backup, view errors.err for errors");
}
else
{
System.out.println("Backup taken");
}
}catch(Exception e)
{
System.out.println(e);
}
}
}
}
```

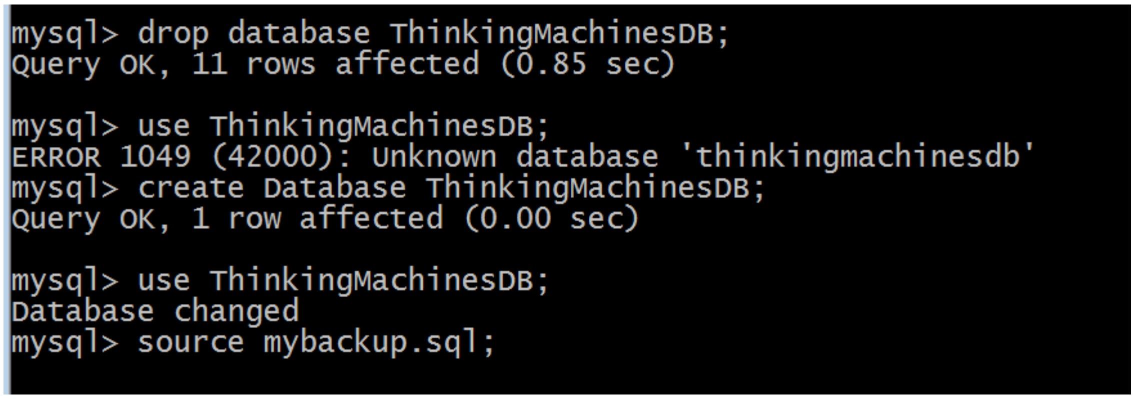
Run the above code, pass file name in which you want to take backup for eg.
java -classpath c:\mysql\*;. jdbc15 mybackup.sql

If everything is correct, then the mybackup.sql file will be created, which can be used later on to restore database.

Ideally in a project, we will generate the backup file name dynamically using date time etc.

Restoring from backup file

I am assuming that you have taken backup in mybackup.sql
First of all let us drop the ThinkingMachinesDB, for that login into mysql (root user) and type
drop database ThinkingMachinesDB;
to verify, type
use ThinkingMachinesDB;
you should get an error message
Now type
create database ThinkingMachinesDB;
then type
using ThinkingMachinesDB;
then type
source mybackup.sql;



```
mysql> drop database ThinkingMachinesDB;
Query OK, 11 rows affected (0.85 sec)

mysql> use ThinkingMachinesDB;
ERROR 1049 (42000): Unknown database 'thinkingmachinesdb'
mysql> create Database ThinkingMachinesDB;
Query OK, 1 row affected (0.00 sec)

mysql> use ThinkingMachinesDB;
Database changed
mysql> source mybackup.sql;
```

Done, your database has been restored, you can exit from mysql, login as tmdbuser and check the restored records.

Multithreading – The traditional way

thread1.java (will not compile)

```
class aaa
{
aaa()
{
Thread t;
t=new Thread(this);
}
}
class thread1
{
public static void main(String gg[])
{
aaa a;
a=new aaa();
int x;
x=1;
while(x<=200)
{
System.out.print(x+" ");
x++;
}
}
}
```

---

thread2.java (will not compile)

```
class aaa implements Runnable
{
aaa()
{
Thread t;
t=new Thread(this);
}
}
class thread2
{
public static void main(String gg[])
{
aaa a;
a=new aaa();
int x;
x=1;
while(x<=200)
{
System.out.print(x+" ");
x++;
}
}
}
```

Note : run this code many times, every time change the load factor on OS (run many apps in parallel)

thread3.java (will compile)

```
class aaa implements Runnable
{
aaa()
{
Thread t;
t=new Thread(this);
t.start(); // the run will be loaded on a separate Thread to which (t) is pointing
}
public void run()
{
for(int j=2001;j<=2200;j++)
{
System.out.print(j+" ");
}
}
}
// When we write java psp, JVM creates a thread and loads the
// entry point function on it
class psp
{
public static void main(String gg[])
{
aaa a;
a=new aaa();
int x;
x=1;
while(x<=200)
{
System.out.print(x+" ");
x++;
}
}
}
```

Run this code many times as discussed earlier

thread4.java (will compile)

```
class aaa extends Thread
{
aaa()
{
start();
}
public void run()
{
for(int j=2001;j<=2200;j++)
{
System.out.print(j+" ");
}
}
}
// When we write java psp, JVM creates a thread and loads the
// entry point function on it
class thread4
{
public static void main(String gg[])
{
aaa a;
a=new aaa();
int x;
x=1;
while(x<=200)
{
System.out.print(x+" ");
x++;
}
}
}
```

Synchronization

Run the code many times as done earlier

thread5.java (will compile)

```
// Problems associated with multi threading
//What will happen when multiple
// threads will work on a common object
class cmn
{
private String m;
public void sam(String g)
{
m=g;
System.out.println(m);
try
{
Thread.sleep(1000); // Thread goes to sleep for 1 second(1000 milliseconds)
}catch(InterruptedException ie) {}
System.out.println(m);
}
}
class worker extends Thread
```

```
{
private cmn cc;
private String ss;

worker(cmn c,String s)
{
cc=c;
ss=s;
start();
}
public void run()
{
cc.sam(ss);
}
}
class thread5
{
public static void main(String gg[])
{
cmn c=new cmn();
worker w1=new worker(c,"Hello");
worker w2=new worker(c,"Boys");
worker w3=new worker(c,"Girls");
}
}
```

Run this code many times as done earlier

thread6.java (will compile)

```
// Solution to the previous problem
class cmn
{
private String m;
synchronized public void sam(String g)
{
m=g;
System.out.println(m);
try{
Thread.sleep(1000); // Thread goes to sleep for 1 second(1000 milliseconds)
}catch(InterruptedException ie)
{
}
System.out.println(m);
}
}
class worker extends Thread
{
private cmn cc;
private String ss;
worker(cmn c,String s)
{
cc=c;
ss=s;
start();
}
public void run()
{
cc.sam(ss);
}
}
class thread6
{
public static void main(String gg[])
{
cmn c=new cmn();
worker w1=new worker(c,"Hello");
worker w2=new worker(c,"Boys");
worker w3=new worker(c,"Girls");
}
}
```

Run this code many times as done earlier

thread7.java (will compile)

```
class cmn
{
private String m;
public void sam(String g)
{
m=g;
System.out.println(m);
try
{
Thread.sleep(1000); // Thread goes to sleep for 1 second(1000 milliseconds)
}catch(InterruptedException ie)
{
}
System.out.println(m);
}
}
class worker extends Thread
{
private cmn cc;
private String ss;
worker(cmn c,String s)
{
```

```
cc=c;
ss=s;
start();
}
public void run()
{
synchronized(cc)
{
cc.sam(ss);
}
}
}
class thread7
{
public static void main(String gg[])
{
cmn c=new cmn();
worker w1=new worker(c,"Hello");
worker w2=new worker(c,"Boys");
worker w3=new worker(c,"Girls");
}
}
```

Classic producer / consumer scenario and synchronization

Run this code many times as done earlier

thread8.java (will compile)

```
class mdm
{
private int num;
public void setNumber(int n)
{
num=n;
System.out.println("Produced : "+num);
}
public int getNumber()
{
System.out.println("Consumed : "+num);
return num;
}
}
class Producer extends Thread
{
private mdm m;
Producer(mdm m)
{
this.m=m;
start();
}
public void run()
{
for(int x=201;x<=250;x++)
{
m.setNumber(x);
}
}
}
class Consumer extends Thread
{
private mdm m;
Consumer(mdm m)
{
this.m=m;
start();
}
public void run()
{
int e,f;
for(e=1;e<=50;e++)
{
f=m.getNumber();
}
}
}
class thread8
{
public static void main(String gg[])
{
mdm m=new mdm();
Producer p=new Producer(m);
Consumer c=new Consumer(m);
}
}
```

Run this code many times as done earlier

thread9.java (will compile)

```
class mdm
{
private int num;
private boolean b=false;
synchronized public void setNumber(int n)
{
if(b==true)
{
try
{
wait();
}catch(InterruptedException ie)
{
}
}
num=n;
System.out.println("Produced : "+num);
b=true;
notify();
}
synchronized public int getNumber()
{
if(b==false)
{
try
{
wait();
}catch(InterruptedException ie)
{
}
}
System.out.println("Consumed : "+num);
b=false;
notify();
return num;
}
}
class Producer extends Thread
{
private mdm m;
Producer(mdm m)
{
this.m=m;
start();
}
public void run()
{
for(int x=201;x<=250;x++)
{
m.setNumber(x);
}
}
}
class Consumer extends Thread
{
private mdm m;
```

```
Consumer(mdm m)
{
this.m=m;
start(); }
public void run()
{
int e,f;
for(e=1;e<=50;e++)
{
f=m.getNumber();
}
}
}
class thread9
{
public static void main(String gg[])
{
mdm m=new mdm();
Producer p=new Producer(m);
Consumer c=new Consumer(m);
}
}
```

Local inner classes

inner1.java (will compile)

```
class aaa
{
private int  x;
aaa(int e)
{
x=e;
}
public void joy()
{
System.out.println("I am joy of class aaa");
}
public void sam()
{
class bbb
{
public void joy()
{
System.out.println("I am joy of local inner class bbb");
}
public void tiger()
{
System.out.println(x);
this.joy();
aaa.this.joy();
}
}
bbb b=new bbb();
b.tiger();
}
}
class inner1
{
public static void main(String gg[])
{
aaa a=new aaa(20);
a.sam();
}
}
```

Inner classes
inner2.java (will compile)

```
class aaa
{
private int  x;

class bbb
{
public void joy()
{
System.out.println("I am joy of inner class bbb");
}
public void tiger()
{
System.out.println(x);
this.joy();
aaa.this.joy();
}
}
aaa(int e)
{
x=e;
}
public void joy()
{
System.out.println("I am joy of class aaa");
}
public void sam()
{
bbb b=new bbb();
b.tiger();
}
public void lion()
{
bbb b=new bbb();
b.tiger();
}
}
class inner2
{
public static void main(String gg[])
{
aaa a=new aaa(20);
a.sam();
a.lion();
}
}
```

---

inner3.java (will not compile)

```
class aaa
{
private int  x;

class bbb
{
public void joy()
{
System.out.println("I am joy of  inner class bbb");
}
public void tiger()
{
System.out.println(x);
this.joy();
aaa.this.joy();
}
}
aaa(int e)
{
x=e;
}
public void joy()
{
System.out.println("I am joy of class aaa");
}
public void sam()
{
bbb b=new bbb();
b.tiger();
}
public void lion()
{
bbb b=new bbb();
b.tiger();
}
}
class inner3
{
public static void main(String gg[])
{
aaa.bbb b=new aaa.bbb();
}
}
```

inner4.java (will not compile)

```
class aaa
{
private int  x;
static class bbb
{
public void joy()
{
System.out.println("I am joy of  inner class bbb");
}
public void tiger()
{
System.out.println(x);
this.joy();
aaa.this.joy();
}
}
aaa(int e)
{
x=e;
}
public void joy()
{
System.out.println("I am joy of class aaa");
}
public void sam()
{
bbb b=new bbb();
b.tiger();
}
public void lion()
{
bbb b=new bbb();
b.tiger();
}
}
class inner4
{
public static void main(String gg[])
{
aaa.bbb b=new aaa.bbb();
}
}
```

---

inner5.java (will compile)

```
class aaa
{
private int  x;
static class bbb
{
public void joy()
{
System.out.println("I am joy of inner class bbb");
}
public void tiger()
{
this.joy();
}
}
aaa(int e)
{
x=e;
}
public void joy()
{
System.out.println("I am joy of class aaa");
}
public void sam()
{
bbb b=new bbb();
b.tiger();
}
public void lion()
{
bbb b=new bbb();
b.tiger();
}
}
```

```
class inner5
{
public static void main(String gg[])
{
aaa.bbb b=new aaa.bbb();
b.tiger();
}
}
```

Anonymous classes

anonymous1.java (will compile)

```
class aaa
{
public void sam()
{
System.out.println("Cool");
}
public void toy()
{
System.out.println("great");
}
}
class anonymous1
{
public static void main(String gg[])
{
aaa a=new aaa(){
public void tom()
{
System.out.println("Really great");
}
};
a.sam();
a.toy();
}
}
```

---

anonymous2.java (will not compile)

```
abstract class aaa
{
abstract public void sam();
}
class anonymous2
{
public static void main(String gg[])
{
aaa a=new aaa(){
public void tiger()
{
System.out.println("Cool");
}
};
}
}
```

---

anonymous3.java (will compile)

```
abstract class aaa
{
abstract public void sam();
}
interface bbb
{
public void lion();
}
class anonymous3
{
public static void main(String gg[])
{
aaa a=new aaa(){
public void tiger()
{
System.out.println("Cool");
}
public void sam()
{
System.out.println("Great");
}
};
bbb b=new bbb(){
public void lion()
{
System.out.println("Really great");
}
};
a.sam();
b.lion();
}
}
```

---

anonymous4.java (will compile)

```
class anonymous4
{
public static void main(String gg[])
{
Runnable r=new Runnable(){
public void run()
{
for(int j=2001;j<=2200;j++)
{
System.out.print(j+" ");
}
}
};
Thread t=new Thread(r);
t.start();
int x;
x=1;
while(x<=200)
{
System.out.print(x+" ");
x++;
}
}
}
```

anonymous5.java (will compile)

```
class anonymous5
{
public static void main(String gg[])
{
Thread t=new Thread(){
public void run()
{
for(int j=2001;j<=2200;j++)
{
System.out.print(j+" ");
}
}
};
t.start();
int x;
x=1;
while(x<=200)
{
System.out.print(x+" ");
x++;
}
}
}
```

Lambda

lambda1.java (will compile)

```
interface Host
{
public void welcome(String name);
}
interface FeeCalculator
{
public int getCourseFee(String course);
}
class lambda1
{
public static void main(String gg[])
{
Host indianHost=(message)-> { System.out.printf("Namaste %s\n",message);};
Host americanHost=(message)-> { System.out.printf("Hello %s\n",message);};
indianHost.welcome("Sameer");
americanHost.welcome("Sameer");
FeeCalculator javaCourse=(course)->{
if(course.equals("Java")) return 10000;
};
}
}
```

---

lambda2.java (will compile)

```
interface calculator
{
public int calculate(int num1,int num2);
}
class lambda2
{
public static void main(String gg[])
{
calculator add=(number1,number2)->number1+number2;
calculator substract=(number1,number2)->number1-number2;
calculator multiply=(number1,number2)->number1*number2;
calculator divide=(number1,number2)->number1/number2;
System.out.printf("Total is %d\n",add.calculate(10,2));
System.out.printf("Difference is %d\n",substract.calculate(10,2));
System.out.printf("Product is is %d\n",multiply.calculate(10,2));
System.out.printf("Quotient is %d\n",divide.calculate(10,2));
}
}
```

---

lambda3.java (will not compile)

```
interface calculator
{
public int calculate(int num1,int num2);
}
class lambda3
{
public static void main(String gg[])
{
calculator add=(number1,number2)-> return number1+number2;
calculator substract=(number1,number2)-> return number1-number2;
calculator multiply=(number1,number2)-> return number1*number2;
calculator divide=(number1,number2)-> return number1/number2;
System.out.printf("Total is %d\n",add.calculate(10,2));
System.out.printf("Difference is %d\n",substract.calculate(10,2));
System.out.printf("Product is is %d\n",multiply.calculate(10,2));
System.out.printf("Quotient is %d\n",divide.calculate(10,2));
}
}
```

lambda4.java (will compile)

```
interface calculator
{
public int calculate(int num1,int num2);
}
class lambda4
{
public static void main(String gg[])
{
calculator add=(number1,number2)-> { return number1+number2; };
calculator substract=(number1,number2)-> { return number1-number2; };
calculator multiply=(number1,number2)-> { return number1*number2; };
calculator divide=(number1,number2)-> { return number1/number2; };
System.out.printf("Total is %d\n",add.calculate(10,2));
System.out.printf("Difference is %d\n",substract.calculate(10,2));
System.out.printf("Product is is %d\n",multiply.calculate(10,2));
System.out.printf("Quotient is %d\n",divide.calculate(10,2));
}
}
```

Multithreading – Concurrency - The new technique.

concurr1.java (will compile)

```
class concurr1
{
public static void main(String gg[])
{
Runnable r=()->{
for(int y=301;y<=350;y++)
{
System.out.print(y+" ");
}
};
Thread t=new Thread(r);
t.start();
for(int x=1;x<=50;x++)
{
System.out.print(x+" ");
}
}
}
```

Note : when you will run the following code, it will get stuck in end, press control C to end application.

Concurrency – ExecutorService

concurr2.java (will compile)

```
import java.util.concurrent.*;
class concurr2
{
public static void main(String g[])
{
ExecutorService es=Executors.newSingleThreadExecutor();
es.submit(()->{
for(int x=1;x<=50;x++)
{
System.out.print(x+" ");
}
});
for(int y=201;y<=250;y++)
{
System.out.print(y+" ");
}
}
}
```

---

concurr3.java (will compile)

```
import java.util.concurrent.*;
class concurr3
{
public static void main(String g[])
{
ExecutorService es=Executors.newSingleThreadExecutor();
es.submit(()->{
for(int x=1;x<=50;x++)
{
System.out.print(x+" ");
}
});
for(int y=201;y<=250;y++)
{
System.out.print(y+" ");
}
es.shutdown();
}
}
```

Note : After running the following code, wait for some time, don't press Control C

Concurrency – Callable interface & Future task
concurr4.java (will compile)

```
import java.util.concurrent.*;
class concurr4
{
public static void main(String gg[])
{
Callable<Integer> work=()->{
TimeUnit.SECONDS.sleep(10);
return 5000;
};
ExecutorService es=Executors.newSingleThreadExecutor();
Future<Integer> future=es.submit(work);
System.out.println(future.isDone());
try
{
System.out.println(future.get());
}catch(Exception ie)
{
System.out.println(ie);
}
System.out.println(future.isDone());
es.shutdown();
}
}
```

Concurrency – Thread Pools
concurr5.java (will compile)

```
import java.util.concurrent.*;
class mdm
{
private String m;
public void sam(String g)
{
m=g;
System.out.println(m);
try
{
Thread.sleep(1000);
}catch(InterruptedException ie)
{
}
System.out.println(m);
}
}
class concurr5
{
public static void main(String gg[])
{
String s1="Hello";
String s2="Boys";
String s3="Girls";
mdm c=new mdm();
ExecutorService es;
es=Executors.newFixedThreadPool(3);
Runnable r1=()->{
synchronized(c) { c.sam(s1); }
};
Runnable r2=()->{
synchronized(c) { c.sam(s2); }
};
Runnable r3=()->{
synchronized(c) { c.sam(s3); }
};
es.submit(r1);
es.submit(r2);
es.submit(r3);
es.shutdown();
}
}
```

Concurrency – Locks
concurr6.java (will compile)

```
import java.util.concurrent.*;
import java.util.concurrent.locks.*;
class mdm
{
private String m;
ReentrantLock lock=new ReentrantLock();
public void sam(String g)
{
lock.lock();
m=g;
System.out.println(m);
try
{
Thread.sleep(1000);
}catch(InterruptedException ie)
{
}
System.out.println(m);
lock.unlock();
}
}
class concurr6
{
public static void main(String gg[])
{
String s1="Hello";
String s2="Boys";
String s3="Girls";
mdm c=new mdm();
ExecutorService es;
es=Executors.newFixedThreadPool(3);
Runnable r1=()->{
 c.sam(s1);
};
Runnable r2=()->{
 c.sam(s2);
};
Runnable r3=()->{
 c.sam(s3);
};
es.submit(r1);
es.submit(r2);
es.submit(r3);
es.shutdown();
}
}
```

Object Serialization / Deserialization

serialize1.java (will compile)

```
import java.io.*;
class Student
{
int rollNumber;
String name;
public void setRollNumber(int rollNumber) { this.rollNumber=rollNumber; }
public int getRollNumber() { return this.rollNumber; }
public void setName(String name) { this.name=name; }
public String getName() { return this.name; }
}
class serialize1
{
public static void main(String gkk[])
{
try
{
Student s1=new Student();
s1.setRollNumber(101);
s1.setName("Sameer");
ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);
oos.writeObject(s1);
oos.flush();
byte bytes[];
bytes=baos.toByteArray();
System.out.println("Object serialized to byte array of length : "+bytes.length);
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

serialize2.java (will compile)

```
import java.io.*;
class Student implements Serializable
{
int rollNumber;
String name;
public void setRollNumber(int rollNumber) { this.rollNumber=rollNumber; }
public int getRollNumber() { return this.rollNumber; }
public void setName(String name) { this.name=name; }
public String getName() { return this.name; }
}
class serialize2
{
public static void main(String gkk[])
{
try
{
Student s1=new Student();
s1.setRollNumber(101);
s1.setName("Sameer");
ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);
oos.writeObject(s1);
oos.flush();
byte bytes[];
bytes=baos.toByteArray();
System.out.println("Object serialized to byte array of length : "+bytes.length);
ByteArrayInputStream bais=new ByteArrayInputStream(bytes);
ObjectInputStream ois=new ObjectInputStream(bais);
Student s2=(Student)ois.readObject();
System.out.println("Byte array data deserialized");
System.out.printf("Roll number %d, Name %s\n",s2.getRollNumber(),s2.getName());
if(s1==s2)
{
System.out.println("Same object");
}
else
{
System.out.println("Another object");
}
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

serialize3.java (will compile)

```
import java.io.*;
class Country implements Serializable
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class State implements Serializable
{
private int code;
private String name;
private Country country;
public void setName(String name)
```

```
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
public void setCountry(Country country)
{
this.country=country;
}
public Country getCountry()
{
return this.country;
}
}
class City implements Serializable
{
private int code;
private String name;
private State state;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
public void setState(State state)
{
```

```
this.state=state;
}
public State getState()
{
return this.state;
}
}
class Category implements Serializable
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class Branch implements Serializable
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
```

```
}
}
class Student implements Serializable
{
private int rollNumber;
private String name;
private City city;
private Branch branch;
private Category category;
private String hobbies[];
private int marks[];
public void setRollNumber(int rollNumber)
{
this.rollNumber=rollNumber;
}
public void setName(String name)
{
this.name=name;
}
public void setCity(City city)
{
this.city=city;
}
public void setBranch(Branch branch)
{
this.branch=branch;
}
public void setCategory(Category category)
{
this.category=category;
}
public void setHobbies(String hobbies[])
{
this.hobbies=hobbies;
}
public void setMarks(int marks[])
{
this.marks=marks;
}
public int getRollNumber()
{
return this.rollNumber;
}
public String getName()
{
return this.name;
}
```

```
public City getCity()
{
return this.city;
}
public Branch getBranch()
{
return this.branch;
}
public Category getCategory()
{
return this.category;
}
public String [] getHobbies()
{
return this.hobbies;
}
public int[] getMarks()
{
return this.marks;
}
}
class serialize3
{
public static void main(String gg[])
{
Country country=new Country();
country.setCode(1);
country.setName("India");
State state=new State();
state.setCode(101);
state.setName("Madhya Pradesh");
state.setCountry(country);
City city=new City();
city.setCode(1001);
city.setName("Ujjain");
city.setState(state);
Branch branch=new Branch();
branch.setCode(5001);
branch.setName("Computer Science");
Category category=new Category();
category.setCode(6001);
category.setName("General");
Student student1=new Student();
student1.setRollNumber(10001);
student1.setName("Sameer Gupta");
student1.setHobbies(new String[]{"Reading fiction","Solving Crossword Puzzles","Robotics"});
student1.setMarks(new int[]{91,93,92,85,93});
student1.setCity(city);
student1.setBranch(branch);
student1.setCategory(category);
try
{
ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);
oos.writeObject(student1);
byte bytes[]=baos.toByteArray();
System.out.println("Object serialized to byte array of length : "+bytes.length);
Student student2;
ByteArrayInputStream bais=new ByteArrayInputStream(bytes);
ObjectInputStream ois=new ObjectInputStream(bais);
student2=(Student)ois.readObject();
System.out.println("Byte array data deserialized");
city=student2.getCity();
state=city.getState();
country=state.getCountry();
branch=student2.getBranch();
category=student2.getCategory();
String []hobbies=student2.getHobbies();
int []marks=student2.getMarks();
System.out.println("Roll number : "+student2.getRollNumber());
System.out.println("Name : "+student2.getName());
System.out.printf("City : Code - %d,  Name - %s\n",city.getCode(),city.getName());
System.out.printf("State : Code - %d,  Name - %s\n",state.getCode(),state.getName());
System.out.printf("Country : Code - %d,  Name - %s\n",country.getCode(),country.getName());
System.out.printf("Branch : Code - %d,  Name - %s\n",branch.getCode(),branch.getName());
System.out.printf("Category : Code - %d,  Name - %s\n",category.getCode(),category.getName());
System.out.println("Hobbies : ");
for(int i=0;i<hobbies.length;i++)
{
System.out.printf("\t %s\n",hobbies[i]);
}
System.out.println("Marks : ");
for(int i=0;i<marks.length;i++)
{
System.out.printf("\t %d\n",marks[i]);
}
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
```

Socket Programming – Introduction

Create a folder named as socket1
in it create socket1.java

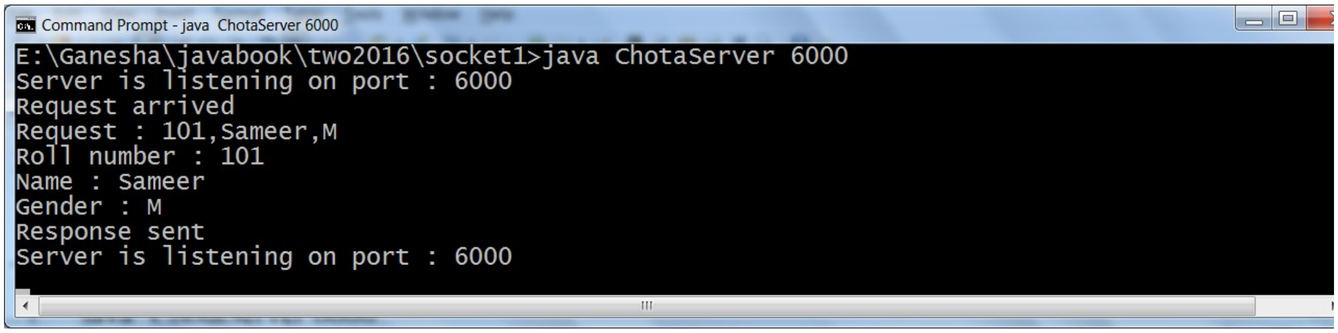
socket1.java (will compile)

```
import java.io.*;
import java.net.*;
class ChotaClient {
public static void main(String data[])
{
String serverName=data[0];
int portNumber=Integer.parseInt(data[1]);
int rollNumber=Integer.parseInt(data[2]);
String name=data[3];
String gender=data[4];
String request=rollNumber+","+name+","+gender+"#";
try
{
Socket socket=new Socket(serverName,portNumber);
OutputStream os;
OutputStreamWriter osw;
InputStream is;
InputStreamReader isr;
StringBuffer sb;
String response;
int x;
os=socket.getOutputStream();
osw=new OutputStreamWriter(os);
osw.write(request);
osw.flush(); // request sent
is=socket.getInputStream();
isr=new InputStreamReader(is);
sb=new StringBuffer();
while(true)
{
x=isr.read();
if(x=='#' || x==-1)
{
break;
}
sb.append((char)x);
}
response=sb.toString();
System.out.println(response);
socket.close();
}catch(Exception e)
{
System.out.println(e);
```

```
}
}

}
class ChotaServer
{
private ServerSocket serverSocket;
private int portNumber;
ChotaServer(int portNumber)
{
this.portNumber=portNumber;
try
{
serverSocket=new ServerSocket(this.portNumber);
startListening();
}catch(Exception e)
{
System.out.println(e);
System.exit(0);
}
}
public void startListening()
{
try
{
InputStream is;
InputStreamReader isr;
OutputStream os;
OutputStreamWriter osw;
StringBuffer sb;
String request;
int x;
int c1,c2;
String pc1,pc2,pc3;
int rollNumber;
String name;
String gender;
Socket ck;
String response;
while(true)
{
System.out.println("Server is listening on port : "+this.portNumber);
ck=serverSocket.accept();
System.out.println("Request arrived ");
is=ck.getInputStream();
isr=new InputStreamReader(is);
sb=new StringBuffer();
while(true)
{
x=isr.read();

if(x=='#' || x==-1)
{
break;
}
sb.append((char)x);
}
request=sb.toString();
System.out.println("Request : "+request);
c1=request.indexOf(",");
c2=request.indexOf(",",c1+1);
pc1=request.substring(0,c1);
pc2=request.substring(c1+1,c2);
pc3=request.substring(c2+1);
rollNumber=Integer.parseInt(pc1);
name=pc2;
gender=pc3;
System.out.println("Roll number : "+rollNumber);
System.out.println("Name : "+name);
System.out.println("Gender : "+gender);
// code to save data
response="Data saved#";
os=ck.getOutputStream();
osw=new OutputStreamWriter(os);
osw.write(response);
osw.flush();
System.out.println("Response sent");
ck.close();
}
}catch(Exception e)
{
System.out.println(e);
}
}
public static void main(String data[])
{
int portNumber=Integer.parseInt(data[0]);
ChotaServer cs=new ChotaServer(portNumber);
}
}
```

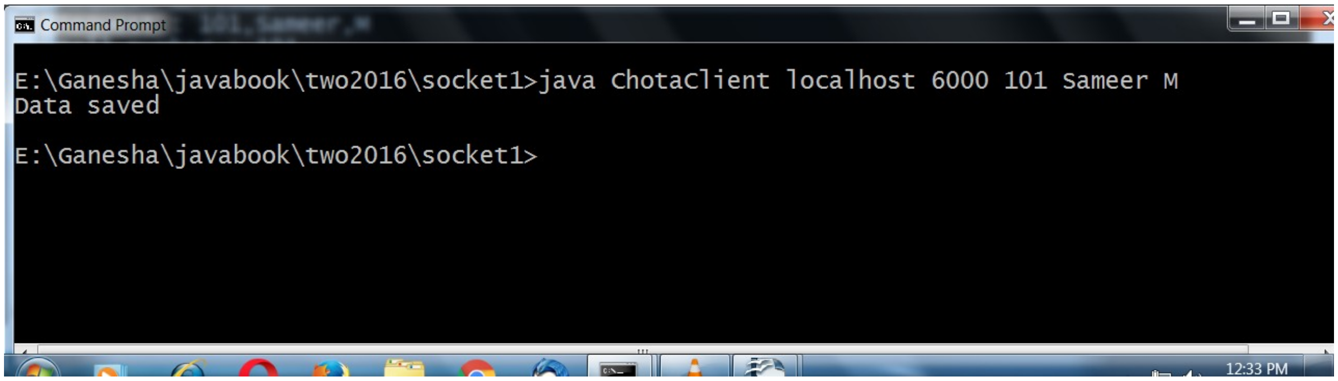
After compiling the above code. Open 2 command windows, resize size them to smaller size and keep them vertically in parallel to each other.
move into the socket1 folder in both of them.
In one of them type
java ChotaServer 6000

Note : if firewall prompts a message dialog (click the allow button)
Now the server will go in listening mode.
In the another window, type
java ChotaClient localhost 6000 101 Sameer M
The client code will complete after sending the request and receiving back the response. The server will be still in listening mode for next request. You can press Control C to terminate the server application.
The server window screen shot



```
E:\Ganesha\javabook\two2016\socket1>java ChotaServer 6000
Server is listening on port : 6000
Request arrived
Request : 101,Sameer,M
Roll number : 101
Name : Sameer
Gender : M
Response sent
Server is listening on port : 6000
```

The Client window screen shot



```
E:\Ganesha\javabook\two2016\socket1>java ChotaClient localhost 6000 101 Sameer M
Data saved

E:\Ganesha\javabook\two2016\socket1>
```

Note : You can connect two machines, note down their IP Addresses, then while running the client application, you can specify the IP Address of the machines running the server application.
I have specified (localhost) as the server is running on the same machine on which the client is running.

Henceforth to do the same to run server/client apps.

Socket programming – Multithreaded server

Create a folder named as socket2
in it create socket2.java

socket2.java (will compile)

```
import java.io.*;
import java.net.*;
class ChotaClient
{
```

```
public static void main(String data[])
{
String serverName=data[0];
int portNumber=Integer.parseInt(data[1]);
int rollNumber=Integer.parseInt(data[2]);
String name=data[3];
String gender=data[4];
String request=rollNumber+","+name+","+gender+"#";
try
{
Socket socket=new Socket(serverName,portNumber);
OutputStream os;
OutputStreamWriter osw;
InputStream is;
InputStreamReader isr;
StringBuffer sb;
String response;
int x;
os=socket.getOutputStream();
osw=new OutputStreamWriter(os);
osw.write(request);
osw.flush(); // request sent
is=socket.getInputStream();
isr=new InputStreamReader(is);
sb=new StringBuffer();
while(true)
{
x=isr.read();
if(x=='#' || x==-1)
{
break;
}
sb.append((char)x);
}
response=sb.toString();
System.out.println(response);
socket.close();
}catch(Exception e)
{
System.out.println(e);
}
}

}
class ChotaServer
{
private ServerSocket serverSocket;
private int portNumber;
ChotaServer(int portNumber)
{
this.portNumber=portNumber;
try
{
serverSocket=new ServerSocket(this.portNumber);
startListening();
}catch(Exception e)
{
System.out.println(e);
System.exit(0);
}
}
public void startListening()
{
try
{
Socket ck;
while(true)
{
System.out.println("Server is listening on port : "+this.portNumber);
ck=serverSocket.accept();
System.out.println("Request arrived ");
new RequestProcessor(ck);
}
}catch(Exception e)
{
System.out.println(e);
}
}
public static void main(String data[])
{
int portNumber=Integer.parseInt(data[0]);
ChotaServer cs=new ChotaServer(portNumber);
}
}
class RequestProcessor extends Thread
{
private Socket ck;
RequestProcessor(Socket socket)
{
this.ck=socket;
start();
}
public void run()
{
```

```
try
{
InputStream is;
InputStreamReader isr;
OutputStream os;
OutputStreamWriter osw;
StringBuffer sb;
String request;
int x;
int c1,c2;
String pc1,pc2,pc3;
int rollNumber;
String name;
String gender;
String response;
is=ck.getInputStream();
isr=new InputStreamReader(is);
sb=new StringBuffer();
while(true)
{
x=isr.read();

if(x=='#' || x==-1)
{
break;
}
sb.append((char)x);
}
request=sb.toString();
System.out.println("Request : "+request);
c1=request.indexOf(",");
c2=request.indexOf(",",c1+1);
pc1=request.substring(0,c1);
pc2=request.substring(c1+1,c2);
pc3=request.substring(c2+1);
rollNumber=Integer.parseInt(pc1);
name=pc2;
gender=pc3;
System.out.println("Roll number : "+rollNumber);
System.out.println("Name : "+name);
System.out.println("Gender : "+gender);
// code to save data
response="Data saved#";
os=ck.getOutputStream();
osw=new OutputStreamWriter(os);
osw.write(response);
osw.flush();
System.out.println("Response sent");
ck.close();
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

run the socket2 code as done in case of socket1

Socket Programming – Sending serialized objects over the network

Create a folder named as socket3
in it create socket3.java

socket3.java (will compile)

```
import java.io.*;
import java.net.*;
class Student implements Serializable
{
private int rollNumber;
private String name;
private String gender;
public void setRollNumber(int rollNumber)
{
this.rollNumber=rollNumber;
}
public int getRollNumber()
{
return this.rollNumber;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
public void setGender(String gender)
{
this.gender=gender;
}
public String getGender()
{
return this.gender;
}
}
class StudentClient
```

```
{
public static void main(String data[])
{
String server=data[0];
int portNumber=Integer.parseInt(data[1]);
int rollNumber=Integer.parseInt(data[2]);
String name=data[3];
String gender=data[4];
Student student=new Student();
student.setRollNumber(rollNumber);
student.setName(name);
student.setGender(gender);
try
{
ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);
oos.writeObject(student);
oos.flush();
byte bytes[]=baos.toByteArray();
Socket socket=new Socket(server,portNumber);
OutputStream outputStream=socket.getOutputStream();
int bufferSize=1024;
int numberOfBytesToWrite;
int i=0;
System.out.println("Sending data....");
while(i<bytes.length)
{
numberOfBytesToWrite=bufferSize;
if(i+bufferSize>bytes.length)
{
numberOfBytesToWrite=bytes.length-i;
}
outputStream.write(bytes,i,numberOfBytesToWrite);
outputStream.flush();
i=i+bufferSize;
}
System.out.println("Data sent.......");
InputStream is=socket.getInputStream();
baos=new ByteArrayOutputStream();
byte b[]=new byte[1024];
int byteCount;
while(true)
{
byteCount=is.read(b);
if(byteCount<0) break;
baos.write(b,0,byteCount);
//   break;          // This line needs to be discussed in classroom session, This implementation has a bug
}
b=baos.toByteArray();
String response=new String(b);
socket.close();
System.out.println(response);
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
class StudentServer
{
private ServerSocket serverSocket;
private int portNumber;
StudentServer(int portNumber)
{
this.portNumber=portNumber;
try
{
serverSocket=new ServerSocket(this.portNumber);
startListening();
}catch(Exception e)
{
System.out.println(e);
System.exit(0);
}
}
public void startListening()
{
try
{
Socket ck;
while(true)
{
System.out.println("Server is listening on port : "+this.portNumber);
ck=serverSocket.accept();
System.out.println("Request arrived ");
new RequestProcessor(ck);
}
}catch(Exception e)
{
System.out.println(e);
}
}
public static void main(String data[])
{
```

```
int portNumber=Integer.parseInt(data[0]);
StudentServer cs=new StudentServer(portNumber);
}
}
class RequestProcessor extends Thread
{
private Socket ck;
RequestProcessor(Socket socket)
{
this.ck=socket;
start();
}
public void run()
{
try
{
InputStream is;
OutputStream os;
is=ck.getInputStream();
ByteArrayOutputStream baos=new ByteArrayOutputStream();
byte b[]=new byte[1024];
int byteCount;
System.out.println("Fetching data....");
while(true)
{
byteCount=is.read(b);
System.out.println("Got : "+byteCount+" bytes");
if(byteCount<0) break;
baos.write(b,0,byteCount);
break;
}
System.out.println("Data fetched, now parsing it");
b=baos.toByteArray();
ByteArrayInputStream bais=new ByteArrayInputStream(b);
ObjectInputStream ois=new ObjectInputStream(bais);
Student student=(Student)ois.readObject();
System.out.println("Roll number : "+student.getRollNumber());
System.out.println("Name : "+student.getName());
System.out.println("Gender : "+student.getGender());
os=ck.getOutputStream();
String response="OK";
//we can change the following code to convert object to byte array
// and then write 1024 at a time
// we will do it later one in our project
byte bytes[]=response.getBytes();
os.write(bytes,0,bytes.length);
os.flush();
System.out.println("Response sent");
ck.close();
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

compile and run the server using (java StudentServer 6000) and the client using (java StudentClient localhost 6000 101 Sameer M)

socket4.java (will compile)

Your assignment is to remove the bug as discussed in the classroom session

```
import java.io.*;
import java.net.*;
class Student implements Serializable
{
private int rollNumber;
private String name;
private String gender;
public void setRollNumber(int rollNumber)
{
this.rollNumber=rollNumber;
}
public int getRollNumber()
{
return this.rollNumber;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
public void setGender(String gender)
{
this.gender=gender;
}
public String getGender()
{
return this.gender;
}
}
class StudentClient
{
```

```
public static void main(String data[])
{
String server=data[0];
int portNumber=Integer.parseInt(data[1]);
int rollNumber=Integer.parseInt(data[2]);
String name=data[3];
String gender=data[4];
Student student=new Student();
student.setRollNumber(rollNumber);
student.setName(name);
student.setGender(gender);
try
{
ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);
oos.writeObject(student);
oos.flush();
byte bytes[]=baos.toByteArray();
int size=bytes.length;
byte header[]=new byte[10];
int k=9;
int s=size;
while(k>=0)
{
header[k]=(byte)(s%10);
s=s/10;
k--;
}
System.out.println("Size : "+s);
for(k=0;k<=9;k++)
{
System.out.print(header[k]+" ");
}
Socket socket=new Socket(server,portNumber);
OutputStream outputStream=socket.getOutputStream();
int bufferSize=1024;
int numberOfBytesToWrite;
int i=0;
System.out.println("Sending data....");
outputStream.write(header,0,10);
outputStream.flush();
while(i<bytes.length)
{
numberOfBytesToWrite=bufferSize;
if(i+bufferSize>bytes.length)
{
numberOfBytesToWrite=bytes.length-i;
}
outputStream.write(bytes,i,numberOfBytesToWrite);
outputStream.flush();
i=i+bufferSize;
}
System.out.println("Data sent.......");
InputStream is=socket.getInputStream();
baos=new ByteArrayOutputStream();
byte b[]=new byte[1024];
int byteCount;
while(true)
{
byteCount=is.read(b);
if(byteCount<0) break;
baos.write(b,0,byteCount);
}
b=baos.toByteArray();
String response=new String(b);
socket.close();
System.out.println(response);
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
class StudentServer
{
private ServerSocket serverSocket;
private int portNumber;
StudentServer(int portNumber)
{
this.portNumber=portNumber;
try
{
serverSocket=new ServerSocket(this.portNumber);
startListening();
}catch(Exception e)
{
System.out.println(e);
System.exit(0);
}
}
public void startListening()
{
try
{
```

```
Socket ck;
while(true)
{
System.out.println("Server is listening on port : "+this.portNumber);
ck=serverSocket.accept();
System.out.println("Request arrived ");
new RequestProcessor(ck);
}
}catch(Exception e)
{
System.out.println(e);
}
}
public static void main(String data[])
{
int portNumber=Integer.parseInt(data[0]);
StudentServer cs=new StudentServer(portNumber);
}
}
class RequestProcessor extends Thread
{
private Socket ck;
RequestProcessor(Socket socket)
{
this.ck=socket;
start();
}
public void run()
{
try
{
byte header[]=new byte[10];
InputStream is;
OutputStream os;
is=ck.getInputStream();
ByteArrayOutputStream baos=new ByteArrayOutputStream();
byte b[]=new byte[1024];
int byteCount;
System.out.println("Fetching data....");
is.read(header);
int contentLength=0;
int e,f;
for(e=0;e<=9;e++)
{
System.out.print(header[e]+" ");
}
e=9;
```

```
f=1;
while(e>=0)
{
contentLength=contentLength+(header[e]*f);
e--;
f=f*10;
}
System.out.println("Content length : "+contentLength);
int bytesRead=0;
while(true)
{
byteCount=is.read(b);
if(byteCount<0) break;
bytesRead+=byteCount;
baos.write(b,0,byteCount);
if(bytesRead==contentLength) break;
}
System.out.println("Data fetched, now parsing it");
b=baos.toByteArray();
ByteArrayInputStream bais=new ByteArrayInputStream(b);
ObjectInputStream ois=new ObjectInputStream(bais);
Student student=(Student)ois.readObject();
System.out.println("Roll number : "+student.getRollNumber());
System.out.println("Name : "+student.getName());
System.out.println("Gender : "+student.getGender());
os=ck.getOutputStream();
String response="OK";
//we can change the following code to convert object to byte array
// and then write 1024 at a time
// we will do it later one in our project
byte bytes[]=response.getBytes();
os.write(bytes,0,bytes.length);
os.flush();
System.out.println("Response sent");
ck.close();
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

Socket Programming – The File Server / Client

Create a folder named as socket5
in socket5 create two folders FTServer and FTClient
Create FTServer.java in socket5\FTServer folder
Create FTClient.java in socket5\FTClient folder
Copy some files (video or whatever) to FTClient folder
For example my FTClient folder has a file named as 10004.mp4

FTServer.java (Will compile)

```
import java.io.*;
import java.net.*;
class FTServer
{
private ServerSocket serverSocket;
private int portNumber;
FTServer(int portNumber)
{
this.portNumber=portNumber;
try
{
serverSocket=new ServerSocket(this.portNumber);
startListening();
}catch(Exception e)
{
System.out.println(e);
System.exit(0);
}
}
public void startListening()
{
try
{
Socket ck;
while(true)
{
System.out.println("Server is listening on port : "+this.portNumber);
ck=serverSocket.accept();
System.out.println("Request arrived ");
new RequestProcessor(ck);
}
}catch(Exception e)
{
System.out.println(e);
}
}
public static void main(String data[])
{
int portNumber=Integer.parseInt(data[0]);
```

```
FTServer cs=new FTServer(portNumber);
}
}
class RequestProcessor extends Thread
{
private Socket ck;
RequestProcessor(Socket socket)
{
this.ck=socket;
start();
}
public void run()
{
try
{
InputStream inputStream=ck.getInputStream();
int headerSize=20;
byte response[]={3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3};
byte header[]=new byte[headerSize];
inputStream.read(header);
OutputStream outputStream=ck.getOutputStream();
outputStream.write(response,0,headerSize);
outputStream.flush();
int lengthOfFileName;
int e,f;
lengthOfFileName=0;
e=headerSize-1;
f=1;
while(e>=0)
{
lengthOfFileName=lengthOfFileName+(header[e]*f);
e--;
f=f*10;
}
System.out.println(lengthOfFileName+"((((");
int bufferSize=1024;
byte bytes[]=new byte[bufferSize];
int byteRead=0;
ByteArrayOutputStream baos=new ByteArrayOutputStream();
int byteCount;
while(true)
{
byteCount=inputStream.read(bytes);
System.out.println(byteCount);
if(byteCount<0) break;
baos.write(bytes,0,byteCount);
byteRead+=byteCount;
if(byteRead==lengthOfFileName) break;
}
System.out.println("Serialized form of file name received");
bytes=baos.toByteArray();
outputStream.write(response,0,headerSize);
outputStream.flush();
String fileName;
ByteArrayInputStream bais=new ByteArrayInputStream(bytes);
ObjectInputStream ois=new ObjectInputStream(bais);
fileName=(String)ois.readObject();
System.out.println("Receiving file : "+fileName);
inputStream.read(header);
outputStream.write(response,0,headerSize);
outputStream.flush();
long lengthOfFile;
lengthOfFile=0;
e=headerSize-1;
f=1;
while(e>=0)
{
lengthOfFile=lengthOfFile+(header[e]*f);
e--;
f=f*10;
}
System.out.println("Length of file : "+lengthOfFile);
File file=new File(fileName);
if(file.exists()) file.delete();
FileOutputStream fileOutputStream;
fileOutputStream=new FileOutputStream(file);
BufferedOutputStream bos=new BufferedOutputStream(fileOutputStream);
bytes=new byte[1024];
int i=0;
int bytesRead;
while(true)
{
bytesRead=inputStream.read(bytes);
if(bytesRead<0) break;
i=i+bytesRead;
bos.write(bytes,0,bytesRead);
bos.flush();
if(i==lengthOfFile) break;
}
outputStream.write(response,0,headerSize);
outputStream.flush();
ck.close();
System.out.println("File received");
}catch(Exception e)
```

```
{
System.out.println(e);
}
}
}
```

Compile the FTServer code and to run type

```
java   FTServer  60000
```

FTClient.java (will compile)

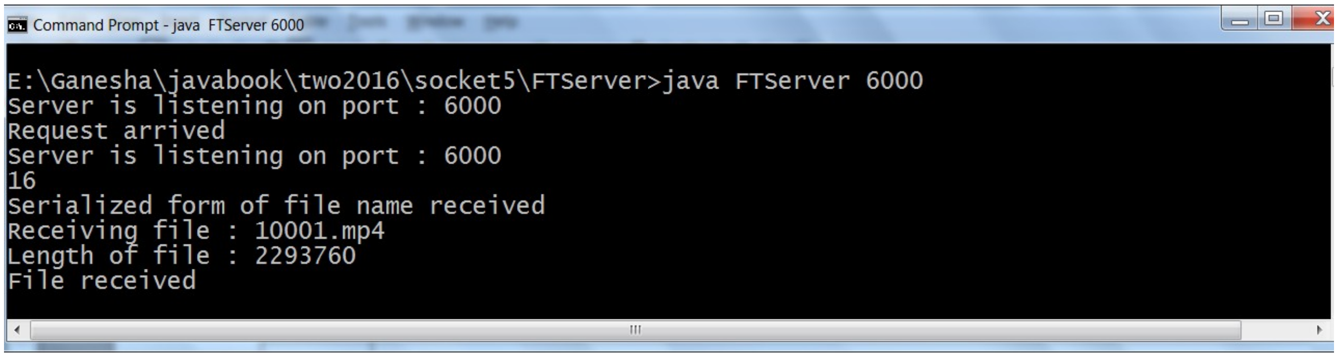
```
import java.net.*;
import java.io.*;
class FTClient
{
public static void main(String data[])
{
try
{
String server=data[0];
int port=Integer.parseInt(data[1]);
String filePath=data[2];
File file=new File(filePath);
if(file.exists()==false)
{
System.out.println("File Not Found : "+filePath);
return;
}
String fileName=file.getName();
ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);
oos.writeObject(fileName);
byte [] fileNameByteArray;
fileNameByteArray=baos.toByteArray();
int lengthOfFileName=fileNameByteArray.length;
int headerSize=20;
byte header[];
header=new byte[headerSize];
int k=headerSize-1;
long f=lengthOfFileName;
while(k>=0)
{
header[k]=(byte)(f%10);
f=f/10;
k--;
}
Socket socket=new Socket(server,port);
OutputStream outputStream;
outputStream=socket.getOutputStream();
```

```
outputStream.write(header,0,headerSize);
outputStream.flush();
byte response[]=new byte[headerSize];
InputStream inputStream=socket.getInputStream();
inputStream.read(response);
int i;
int bufferSize=1024;
int numberOfBytesToWrite=bufferSize;
i=0;
while(i<fileNameByteArray.length)
{
if(i+bufferSize>fileNameByteArray.length)
{
numberOfBytesToWrite=fileNameByteArray.length-i;
}
outputStream.write(fileNameByteArray,i,numberOfBytesToWrite);
outputStream.flush();
inputStream.read(header);
i=i+bufferSize;
}
long lengthOfFile=file.length();
k=headerSize-1;
f=lengthOfFile;
while(k>=0)
{
header[k]=(byte)(f%10);
f=f/10;
k--;
}
outputStream.write(header,0,headerSize);
outputStream.flush();
inputStream.read(response);
System.out.println("header with length of file sent "+lengthOfFile);
FileInputStream fileInputStream;
fileInputStream=new FileInputStream(file);
BufferedInputStream bis=new BufferedInputStream(fileInputStream);
byte contents[]=new byte[1024];
int bytesRead;
i=0;
while(i<lengthOfFile)
{
bytesRead=bis.read(contents);
if(bytesRead<0) break;
outputStream.write(contents,0,bytesRead);
outputStream.flush();
i=i+bytesRead;
}
```

```
fileInputStream.close();
System.out.println("bytes of file sent : "+i);
inputStream.read(response);
// some more code required over here to parse the response
socket.close();
System.out.printf("File sent");
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
```

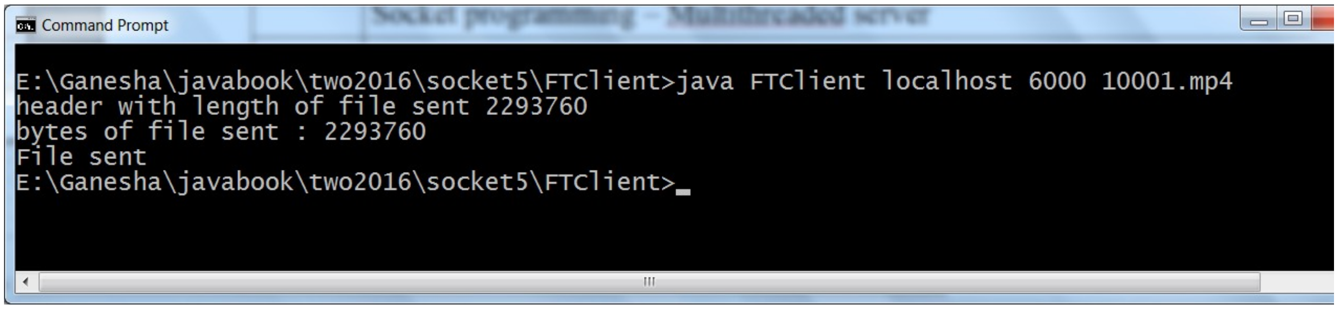
Compile the FTClient code, and to run type
java FTClient localhost 60000 filename
in my case I typed the filename as 10004.mp4

Then in the FTServer folder check the existence of the file that you transferred
The FTServer



```
E:\Ganesha\javabook\two2016\socket5\FTServer>java FTServer 6000
Server is listening on port : 6000
Request arrived
Server is listening on port : 6000
16
Serialized form of file name received
Receiving file : 10001.mp4
Length of file : 2293760
File received
```

TheFTClient



```
E:\Ganesha\javabook\two2016\socket5\FTClient>java FTClient localhost 6000 10001.mp4
header with length of file sent 2293760
bytes of file sent : 2293760
File sent
E:\Ganesha\javabook\two2016\socket5\FTClient>
```

Try sending huge files and see what happens.
Note : Our FTServer / FTClient implementations are not yet final,
We need to optimize our code, we need to write our own protocol implementation and we need to add events as discussed in the classroom session.

Remote Method Invocation (One Way)

Create a folder named as rmi1, in it create rmi1.java

rmi1.java (will compile)

```
import java.rmi.*;
import java.io.*;
import java.rmi.server.*;
class Country implements Serializable
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class State implements Serializable
{
private int code;
private String name;
private Country country;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
```

```
}
public void setCountry(Country country)
{
this.country=country;
}
public Country getCountry()
{
return this.country;
}
}
class City implements Serializable
{
private int code;
private String name;
private State state;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
public void setState(State state)
{
this.state=state;
}
public State getState()
{
return this.state;
}
}
class Category implements Serializable
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
```

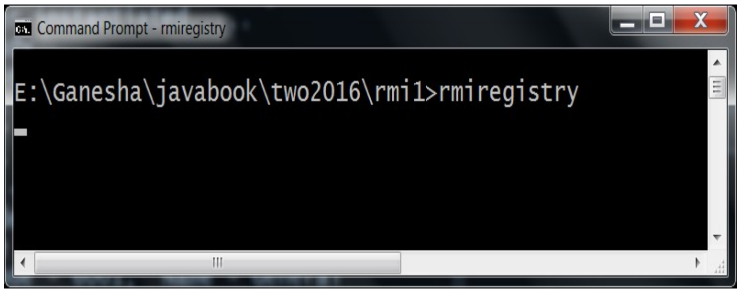
```
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class Branch implements Serializable
{
private int code;
private String name;
public void setName(String name)
{
this.name=name;
}
public void setCode(int code)
{
this.code=code;
}
public String getName()
{
return this.name;
}
public int getCode()
{
return this.code;
}
}
class Student implements Serializable
{
private int rollNumber;
private String name;
private City city;
private Branch branch;
private Category category;
private String hobbies[];
private int marks[];
public void setRollNumber(int rollNumber)
{
this.rollNumber=rollNumber;
```

```
}
public void setName(String name)
{
this.name=name;
}
public void setCity(City city)
{
this.city=city;
}
public void setBranch(Branch branch)
{
this.branch=branch;
}
public void setCategory(Category category)
{
this.category=category;
}
public void setHobbies(String hobbies[])
{
this.hobbies=hobbies;
}
public void setMarks(int marks[])
{
this.marks=marks;
}
public int getRollNumber()
{
return this.rollNumber;
}
public String getName()
{
return this.name;
}
public City getCity()
{
return this.city;
}
public Branch getBranch()
{
return this.branch;
}
public Category getCategory()
{
return this.category;
}
public String [] getHobbies()
{
```

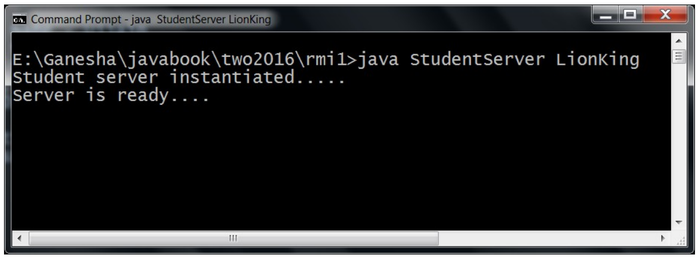
```
return this.hobbies;
}
public int[] getMarks()
{
return this.marks;
}
}
interface StudentServerInterface extends Remote
{
public void addStudent(Student student) throws RemoteException;
public int getNumberOfStudents() throws RemoteException;
}
class StudentServer extends UnicastRemoteObject implements StudentServerInterface
{
StudentServer() throws RemoteException
{
System.out.println("Student server instantiated.....");
}
public void addStudent(Student student) throws RemoteException
{
System.out.println("Request arrived");
City city=student.getCity();
State state=city.getState();
Country country=state.getCountry();
Branch branch=student.getBranch();
Category category=student.getCategory();
String []hobbies=student.getHobbies();
int []marks=student.getMarks();
System.out.println("Roll number : "+student.getRollNumber());
System.out.println("Name : "+student.getName());
System.out.printf("City : Code - %d,  Name - %s\n",city.getCode(),city.getName());
System.out.printf("State : Code - %d,  Name - %s\n",state.getCode(),state.getName());
System.out.printf("Country : Code - %d,  Name - %s\n",country.getCode(),country.getName());
System.out.printf("Branch : Code - %d,  Name - %s\n",branch.getCode(),branch.getName());
System.out.printf("Category : Code - %d,  Name - %s\n",category.getCode(),category.getName());
System.out.println("Hobbies : ");
for(int i=0;i<hobbies.length;i++)
{
System.out.printf("\t %s\n",hobbies[i]);
}
System.out.println("Marks : ");
for(int i=0;i<marks.length;i++)
{
System.out.printf("\t %d\n",marks[i]);
}
}
public int getNumberOfStudents() throws RemoteException
```

```
{
return 605;
}
public static void main(String data[])
{
try
{
String name=data[0];
StudentServer ss=new StudentServer();
Naming.bind(name,ss);
System.out.println("Server is ready....");
}catch(Exception exception)
{
System.out.println(exception);
}
}
}
class StudentClient
{
public static void main(String data[])
{
try
{
String server=data[0];
String serverName=data[1];
StudentServerInterface ssi;
ssi=(StudentServerInterface)Naming.lookup("rmi://"+server+"/"+serverName);


Country country=new Country();
country.setCode(1);
country.setName("India");
State state=new State();
state.setCode(101);
state.setName("Madhya Pradesh");
state.setCountry(country);
City city=new City();
city.setCode(1001);
city.setName("Ujjain");
city.setState(state);
Branch branch=new Branch();
branch.setCode(5001);
branch.setName("Computer Science");
Category category=new Category();
category.setCode(6001);
category.setName("General");
Student student=new Student();
student.setRollNumber(10001);
student.setName("Sameer Gupta");
student.setHobbies(new String[]{"Reading fiction","Solving Crossword Puzzles","Robotics"});
student.setMarks(new int[]{91,93,92,85,93});
student.setCity(city);
student.setBranch(branch);
student.setCategory(category);
ssi.addStudent(student);
System.out.println("Number of students : "+ssi.getNumberOfStudents());
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

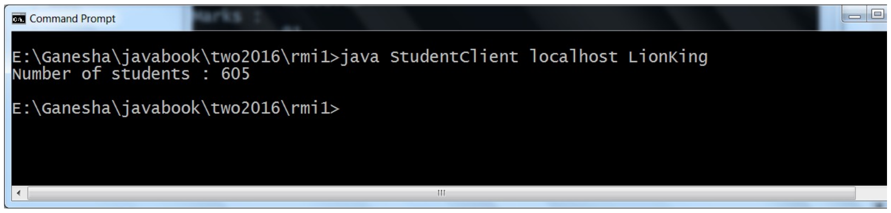
To compile the above code
javac rmi1.java
then (not necessary in case of jdk1.8, but still do it, you may get some warnings, ignore them)
rmic StudentServer
Then open 3 command windows,
move into the rmi1 folder (in all 3 of them)
in first one type (rmiregistry)
in second one type (java StudentServer LionKing)
in third one type (java StudentClient localhost LionKing)



```
E:\Ganesha\javabook\two2016\rmi1>rmiregistry
```

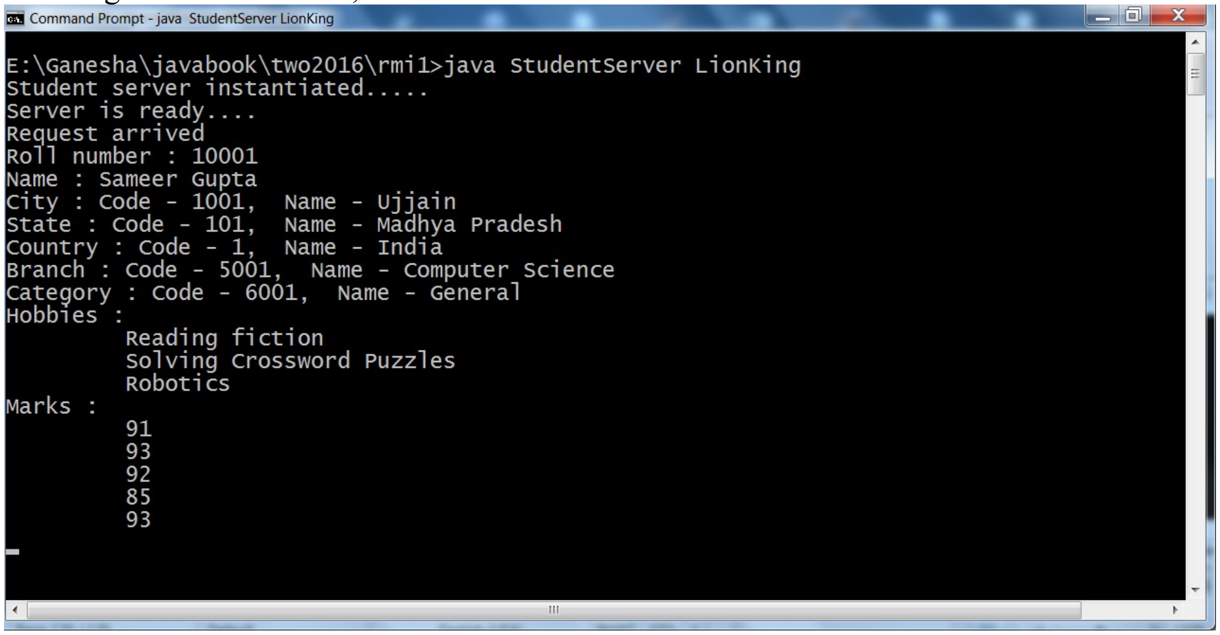


```
E:\Ganesha\javabook\two2016\rmi1>java StudentServer LionKing
Student server instantiated.....
Server is ready....
```



```
E:\Ganesha\javabook\two2016\rmi1>java StudentClient localhost LionKing
Number of students : 605

E:\Ganesha\javabook\two2016\rmi1>
```

After running the StudentClient, the ui of the StudentServer window



```
E:\Ganesha\javabook\two2016\rmi1>java StudentServer LionKing
Student server instantiated.....
Server is ready....
Request arrived
Roll number : 10001
Name : Sameer Gupta
City : Code - 1001,  Name - Ujjain
State : Code - 101,  Name - Madhya Pradesh
Country : Code - 1,  Name - India
Branch : Code - 5001,  Name - Computer Science
Category : Code - 6001,  Name - General
Hobbies :
         Reading fiction
         Solving Crossword Puzzles
         Robotics
Marks :
         91
         93
         92
         85
         93
```

You can end (rmiregistry) and (StudentServer) by pressing Control C