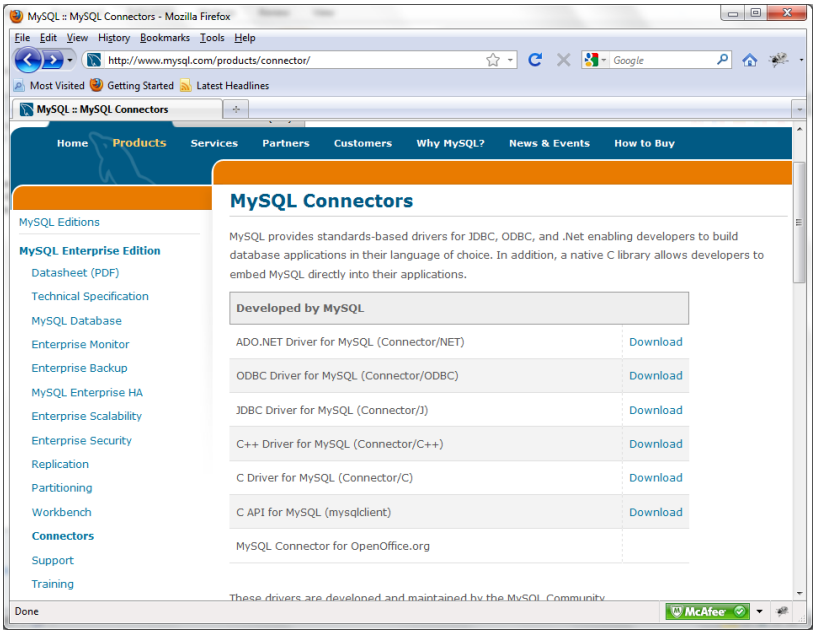
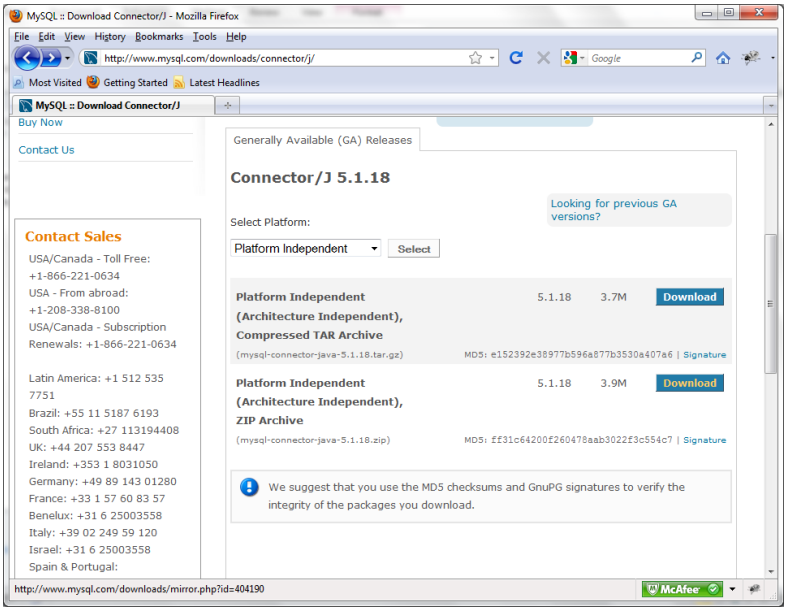
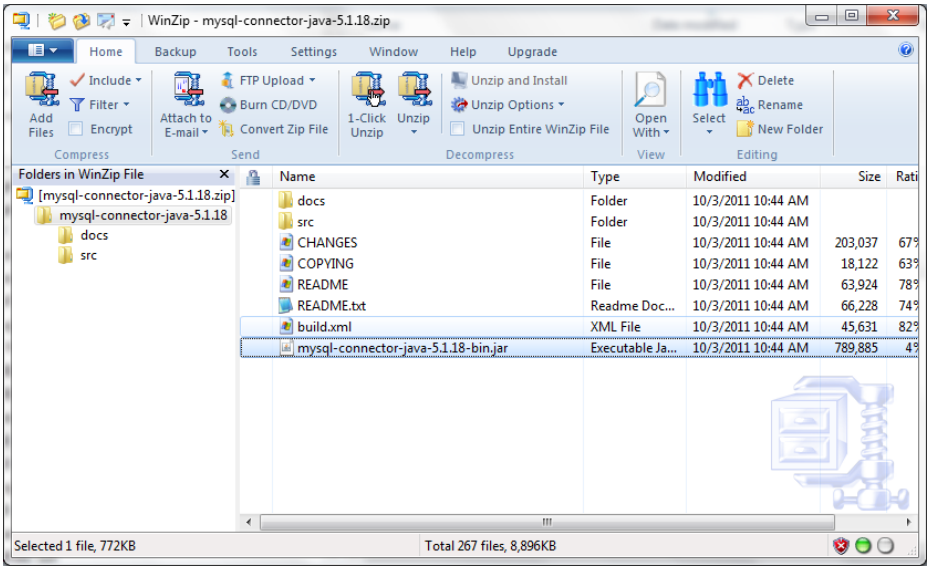
Connecting to MySQL through our Java code

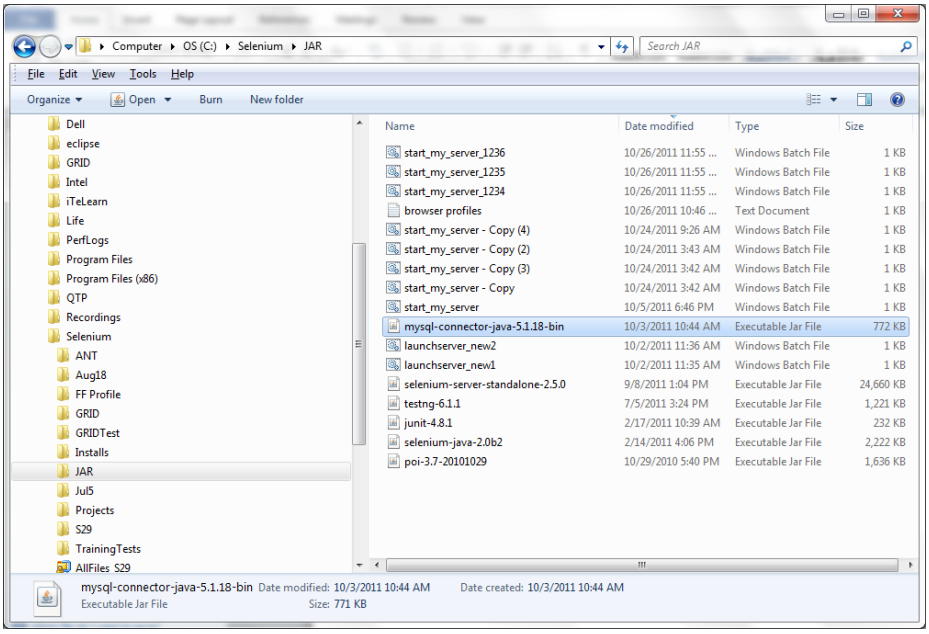
1. Download the Java/JDBC connector jar file for MySQL. Go to http://www.mysql.com/products/connector/





2. Unzip the file and put the jar file into your Jar folder.





3. Parameterizing our Java code
    a. Identify what may be needed.
    b. Define and Initialize variables/parameters.

```
// Define variables
String mySQL, myDB, mySQLDB, myDriver,myT1, myUName, myPswd;
String myQuery1, myQuery2;

Connection myCon = null;
Statement mySt;
PreparedStatementmyPst;
ResultSetrs1, rs2;
```

4. Connecting to Database using JDBC

   Good doc reference http://dev.mysql.com/doc/refman/5.1/en/connector-j-reference.html

```
jdbc:mysql://[host:port],[host:port].../[database] »
     // Initialize variable
     mySQL = "jdbc:mysql://localhost:3306/";
     myDB = "dlearn";
     mySQLDB = mySQL + myDB;
     myDriver = "com.mysql.jdbc.Driver";
     myUName = "root";
     myPswd = "12345";

     // Connecting to DB
     Object myDC = Class.forName(myDriver).newInstance();
     myCon = DriverManager.getConnection(mySQLDB, myUName, myPswd);
mySt = myCon.createStatement();
```

5. Running a sql query on the database using the connection

```
myQuery1 = "Select * from tcourse";
rs1 = mySt.executeQuery(myQuery1);
if (rs1 == null) {
      System.out.print("Empty result set");
}
rs1.next();
System.out.print(rs1.getInt("id"));
System.out.print(rs1.getString("name"));
System.out.print(rs1.getString("cost"));

//Going through all the records
While (rs1.next()) {
System.out.print(rs1.getInt("id"));
System.out.print(rs1.getString("name"));
System.out.print(rs1.getString("cost"));
System.out.println();
}


// Prepared statement for parameterizing our queries
myQuery1 = "Select * from tcourse where cost=? and name=?";
PreparedStatementmyPst = myCon.prepareStatement(myQuery1);
myPst.setString(1, "350");
myPst.setString(2, "Selenium");
rs1 = myPst.executeQuery();
//Going through all the records
rs1.beforeFirst(); // starts from the beginning
while (rs1.next()) {
      if (rs1.isFirst()){
            System.out.println("This is the first");
```

```
        }
        System.out.print(rs1.getInt("id"));
        System.out.print(rs1.getString("name"));
        System.out.print(rs1.getString("cost"));
        System.out.println();
        if (rs1.isLast()){
              System.out.println("This is the Last");
        }
}
```

6. Update records,

```
// Update existing records in the database
System.out.println("Updating existing records");
mySt = myCon.createStatement();
myQuery1 = "update tCourse set cost='475' where id=6";
mySt.executeUpdate(myQuery1);
```

7. Add Records,

```
// Add new records to the database
System.out.println("Add new records");
mySt = myCon.createStatement();
myQuery1 = "INSERT INTO tCourse(id,name,cost)VALUES(6,'Sel++','425')";
mySt.executeUpdate(myQuery1);
// Look in the database for changes now
```

8. Delete Records

```
        // Delete existing records in the database
        System.out.println("Delete existing records");
        mySt = myCon.createStatement();
        myQuery1 = "delete from tCourse where id=6";
        mySt.executeUpdate(myQuery1);
```