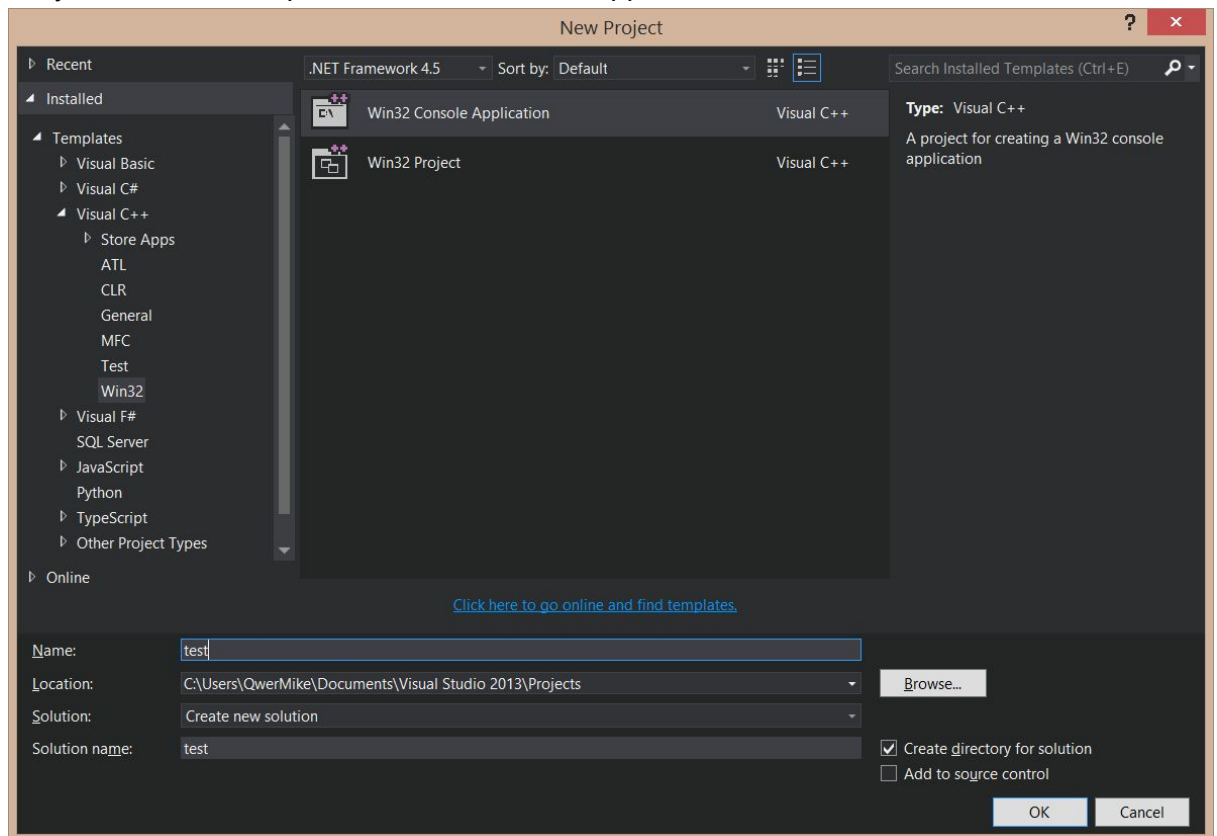
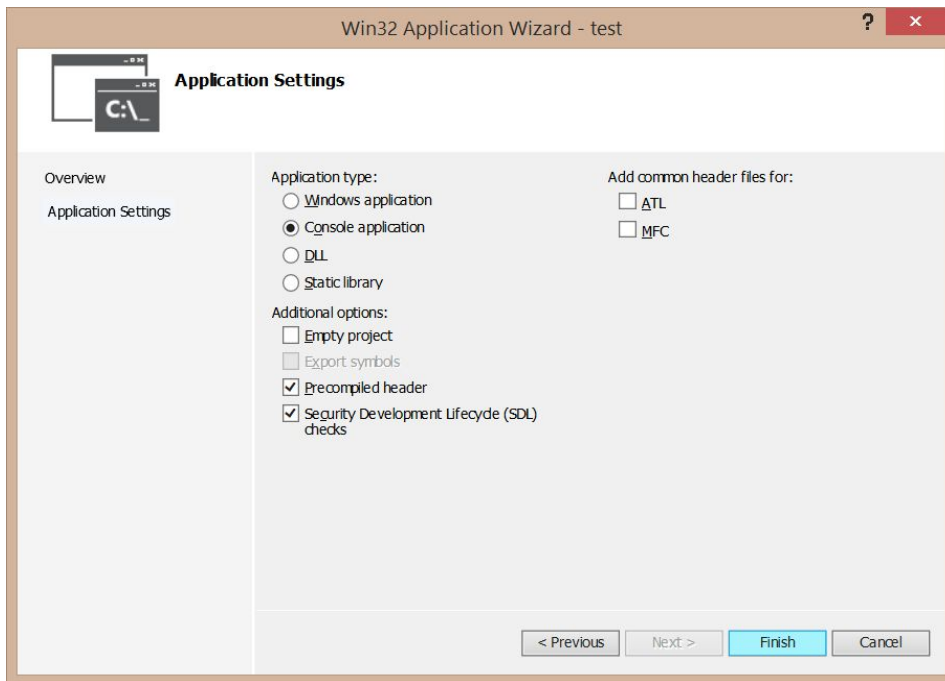


## Інструкція для підключення MySQL до Visual Studio

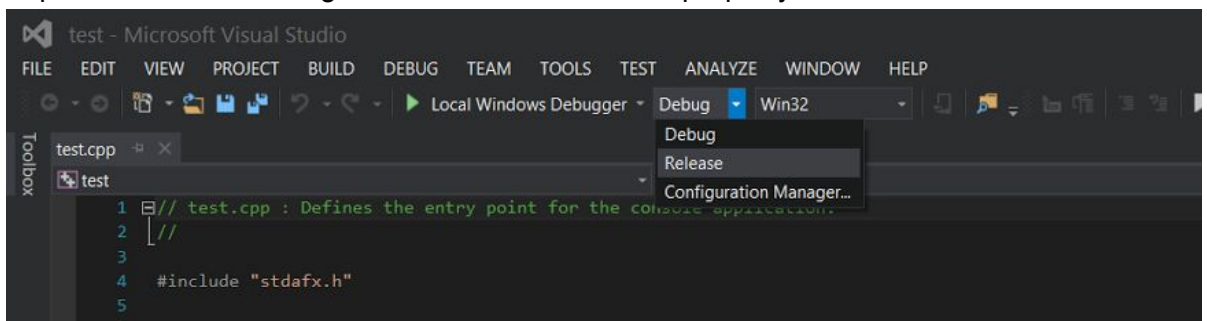
0. Качаємо [MySQL Installer](#), встановлюємо, запускаємо.
1. У MySQL Installer вибираємо метод встановлення - Custom. Вибираємо MySQL Server версію x86 ! та MySQL Connectors -> Connector/C++ версію x86 !!! Тиснемо Next, придумуєм паролі і т. д. Finish.
2. Качаємо додатково бібліотеку [boost](#) і розархівовуємо (зручно в папці з MySQL).
3. Запускаємо VS, створюємо Win32 Console Application. OK.



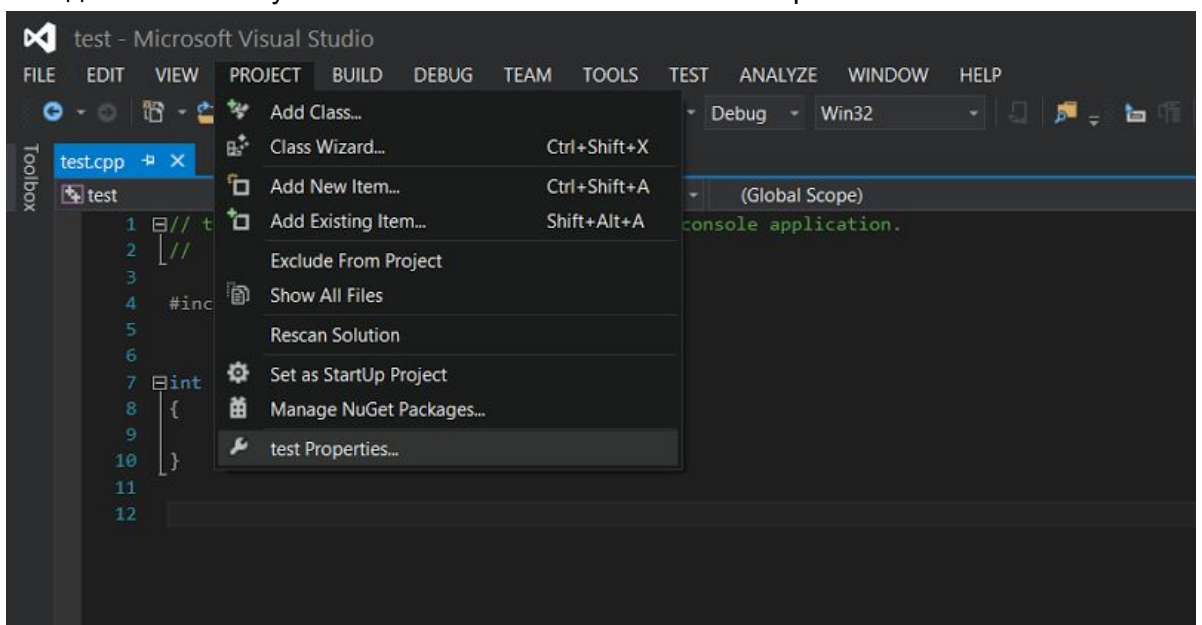
4. Натискаємо Next(краще вибрати empty project тоді `//#include "stdafx.h"` можна викинути з коду)і потім Finish.



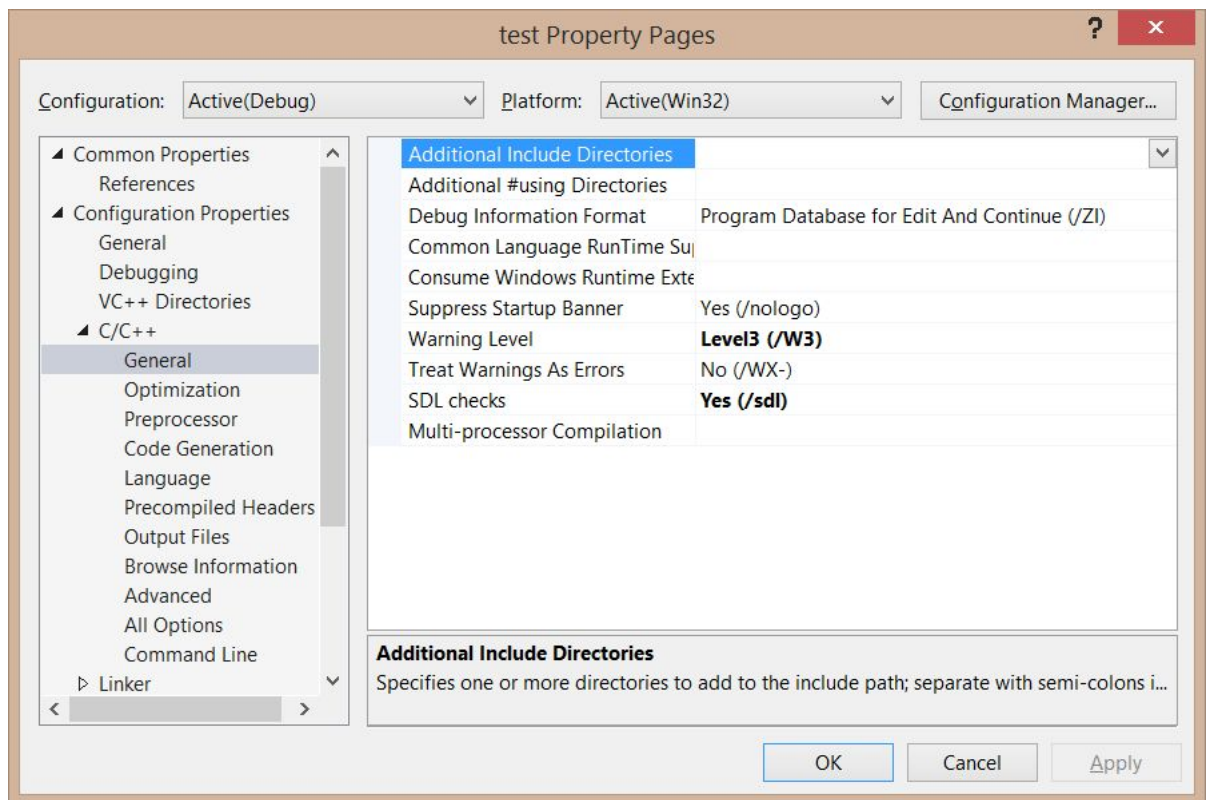
5. Переключаємося з Debug в Release. Компілюємо програму!!!



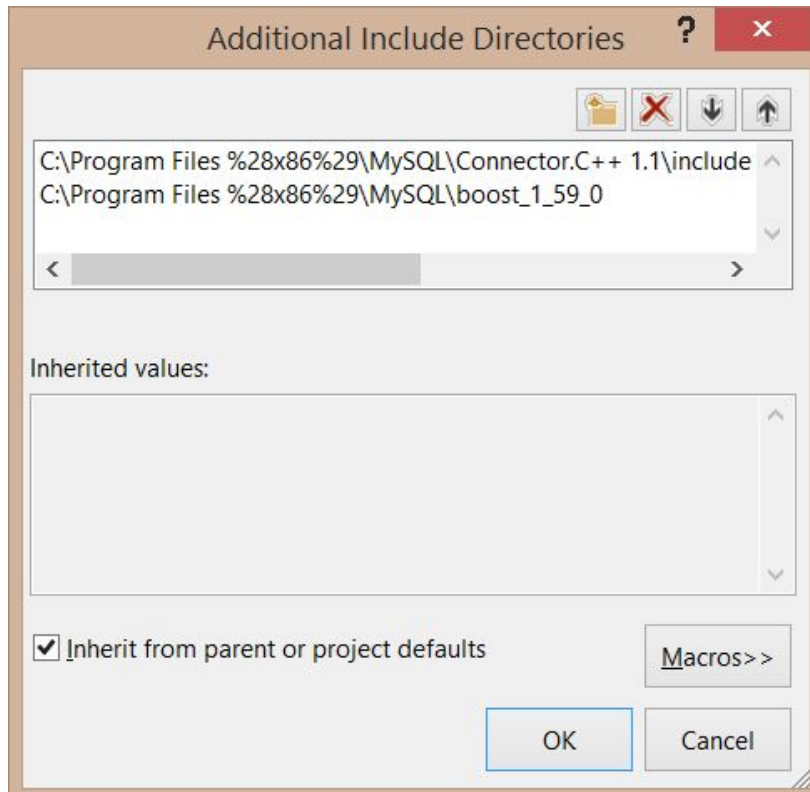
6. Заходимо в налаштування: меню->PROJECT-> 'name' Properties...



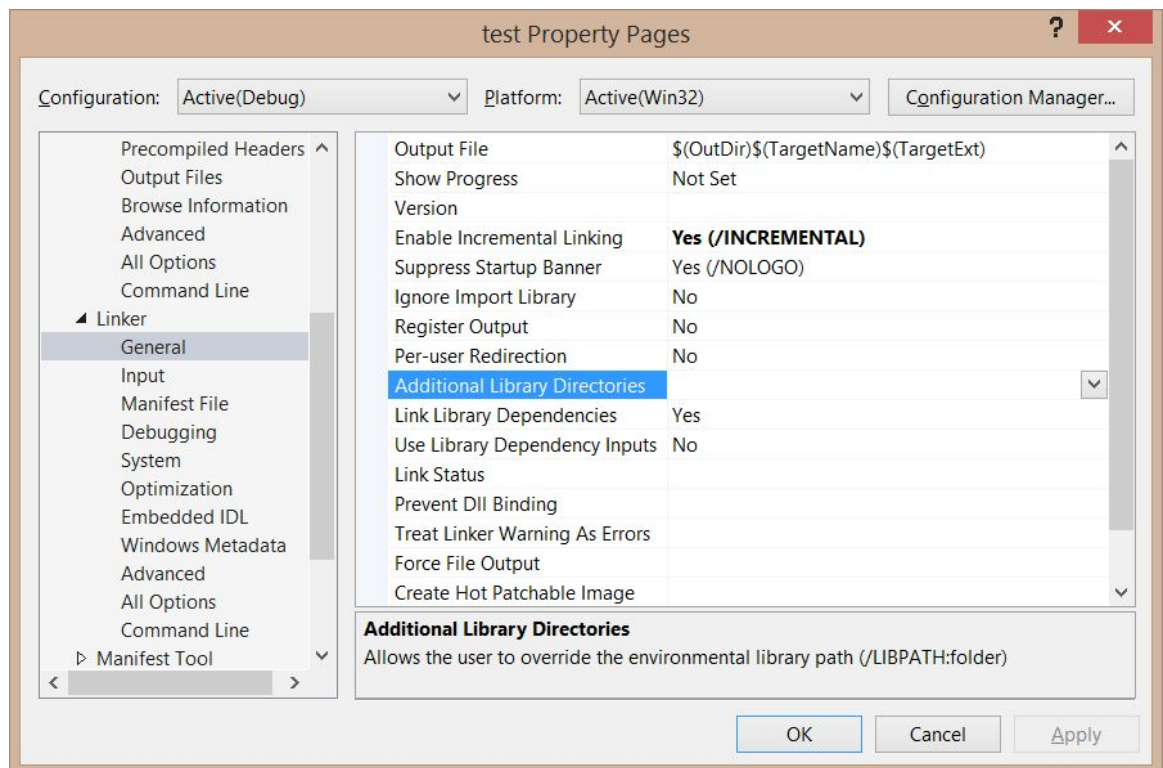
7. Переходимо до Configuration Properties -> C/C++ -> General, тоді справа у вікні Additional Include Directories. <edit>



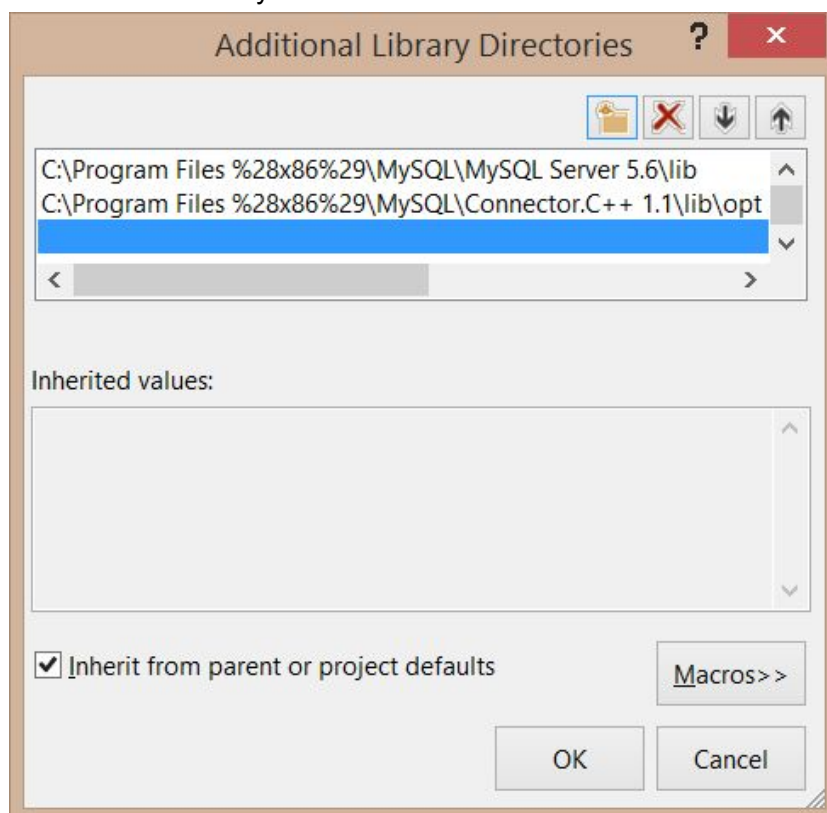
8. Вносимо шляхи \*\\Connector.C++1.1\\include та \*\\boost\_1\_59\_0.



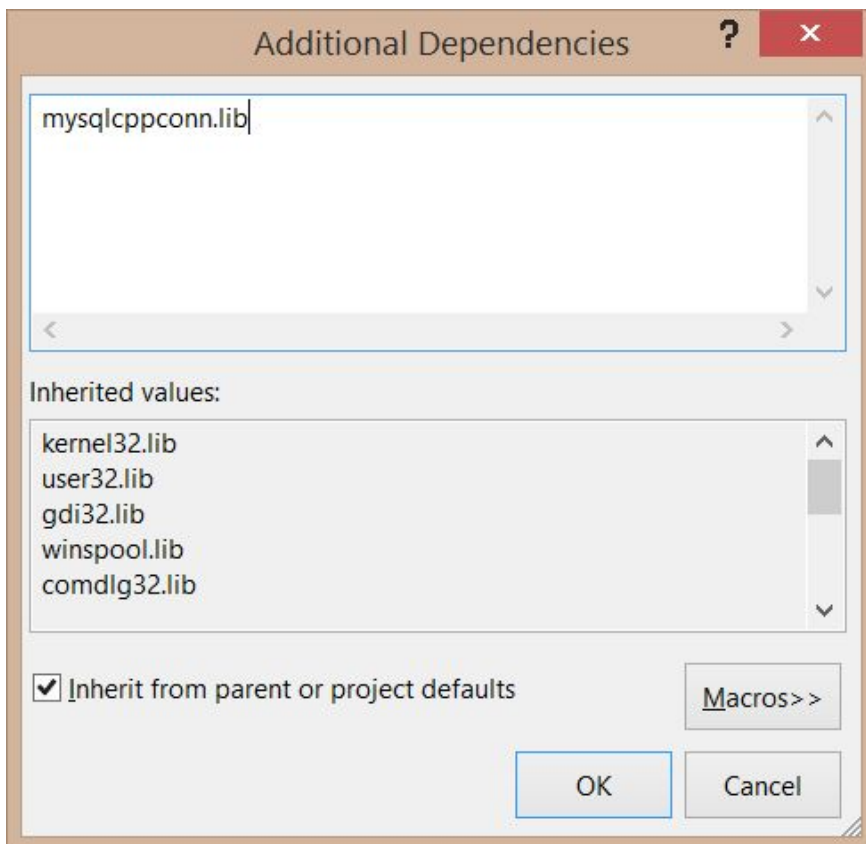
9. Далі відкриваємо Configuration Properties -> Linker -> General. Additional Include Directories. <edit>.



10. Вносимо шлях \*MySQL Server 5.6\lib та \*MySQL Connector.C++ 1.1\lib\opt.



11. Тоді в Linker->Input: Additional Dependencies додаємо текст **mysqlcppconn.lib**



12. Копіюємо файл \*\\Connector.C++ 1.1\\lib\\opt\\mysqlcppconn.dll у папку програми(а саме <ProjectName> / Release /), яка може не існувати, якщо ви не компілювали програму на початку. Можна скопіювати файл у C:/Windows/ замість цього.

13. Копіастим наступний код, компілюємо .

```
// test_SQL.cpp : Defines the entry point for the console application.
```

```
//
```

```
//if u need stdafx.h uncomment next line
```

```
//#include "stdafx.h"
```

```
#include <stdlib.h>
```

```
#include <iostream>
```

```
#include <sstream>
```

```
#include <stdexcept>
```

```
#include "mysql_connection.h"
```

```
#include <cppconn/driver.h>
```

```
#include <cppconn/exception.h>
```

```
#include <cppconn/resultset.h>
```

```
#include <cppconn/statement.h>
```

```
#include <cppconn/prepared_statement.h>
```

```

#define EXAMPLE_HOST "localhost"
#define EXAMPLE_USER "root"
#define EXAMPLE_PASS "aaaaa"
#define EXAMPLE_DB "test"

using namespace std;

int main(int argc, const char **argv)
{
    string url(argc >= 2 ? argv[1] : EXAMPLE_HOST);
    const string user(argc >= 3 ? argv[2] : EXAMPLE_USER);
    const string pass(argc >= 4 ? argv[3] : EXAMPLE_PASS);
    const string database(argc >= 5 ? argv[4] : EXAMPLE_DB);

    try {

        sql::Driver* driver = get_driver_instance();
        std::auto_ptr<sql::Connection> con(driver->connect(url, user, pass));
        con->setSchema(database);
        std::auto_ptr<sql::Statement> stmt(con->createStatement());

    }
    catch (sql::SQLException &e) {
        /*
        MySQL Connector/C++ throws three different exceptions:

        - sql::MethodNotImplementedException (derived from sql::SQLException)
        - sql::InvalidArgumentException (derived from sql::SQLException)
        - sql::SQLException (derived from std::runtime_error)
        */
        cout << "# ERR: SQLException in " << __FILE__;
        cout << "(" << __FUNCTION__ << ") on line " << __LINE__ << endl;
        /* what() (derived from std::runtime_error) fetches error message */
        cout << "# ERR: " << e.what();
        cout << " (MySQL error code: " << e.getErrorCode();
        cout << ", SQLState: " << e.getSQLState() << " )" << endl;

        return EXIT_FAILURE;
    }
    return EXIT_SUCCESS;
}

```