Database Application Development

C++ quick tutorial for the project

C++ TOPICS TO GO THROUGH for the project

- INPUT AND OUTPUT IN C++
- CREATING STRUCTURES
- CREATING FUNCTIONS
- USING OCCI TO CONNECT TO ORACLE

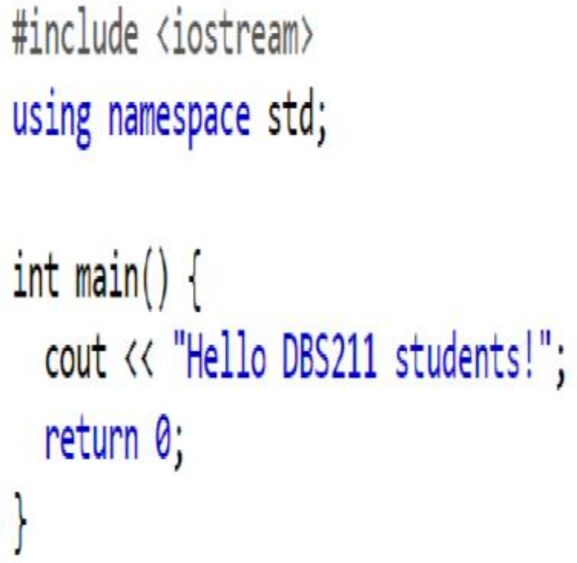
Using standard library header files

#include <iostream>, instructs the preprocessor to include a section of standard C++ code, known as header iostream, that allows to perform standard input and output operations, such as writing the output of this program (Hello DBS211 students!) to the screen.

The function named **main** is a special function in all C++ programs; it is the function called when the program is run. The execution of all C++ programs begins with the main function, regardless of where the function is actually located within the code.

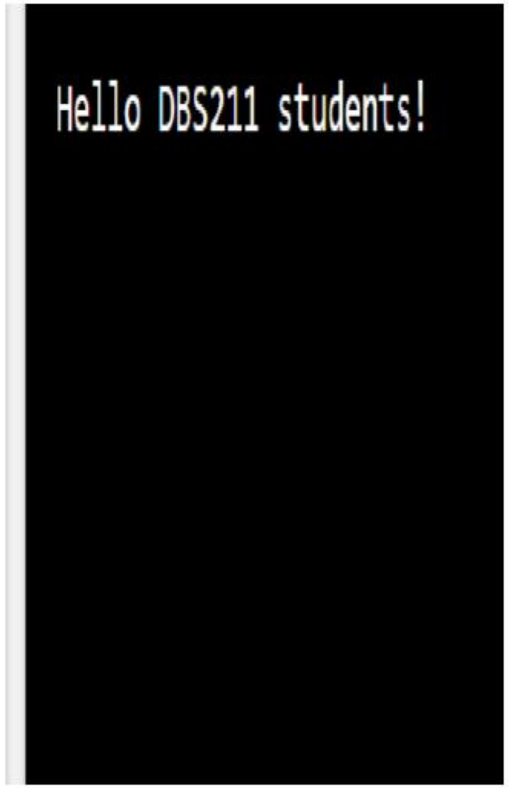
cout is part of the standard library, and all the elements in the standard C++ library are declared within what is called a **namespace:** the namespace std.

Using standard library header files



```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello DBS211 students!";
  return 0;
}
```



```
Hello DBS211 students!
```

Using OCCI header files

Oracle C++ Call Interface (OCCI) is an Application Programming Interface (API) that provides C++ applications access to data in an Oracle database. OCCI enables C++ programmers to use the full range of Oracle database operations, including SQL statement processing and object manipulation.

The Environment class provides an OCCI environment to manage memory and other resources for OCCI objects.
OCCI provides **a library of standard database access and retrieval functions** in the form of a dynamic runtime library (OCCI classes) that can be linked in a C++ application at runtime.

```
#include <occi.h>

using oracle::occi::Environment;
using oracle::occi::Connection;
using namespace oracle::occi;
```

Creating an Environment

```
Environment *env = Environment::createEnvironment();
```

All OCCI objects created with the create*xxx*() methods (connections, connection pools, statements) must be explicitly terminated. When appropriate, you must also explicitly terminate the environment.

Terminating an Environment

If the application requires access to objects in the global scope, such as static or global variables, these objects must be set to NULL before the environment is terminated.

```
Environment::terminateEnvironment(env);
```

createConnection()

This method establishes a connection to the database specified.

Syntax

```
Connection * createConnection(const string &username, const string &password, const string &connectString);

terminateConnection(conn);
```

```
try{
    env=Environment::createEnvironment
             (Environment::DEFAULT);
    conn=env->createConnection(usr,pass,srv);
    cout<<"Connection is Successful!"<<endl;
    env->terminateConnection(conn);
    Environment::terminateEnvironment(env);
}
catch(SQLException& sqlExcp){
    cout<<sqlExcp.getErrorCode()<<
         ":"<<sqlExcp.getMessage();
}
```

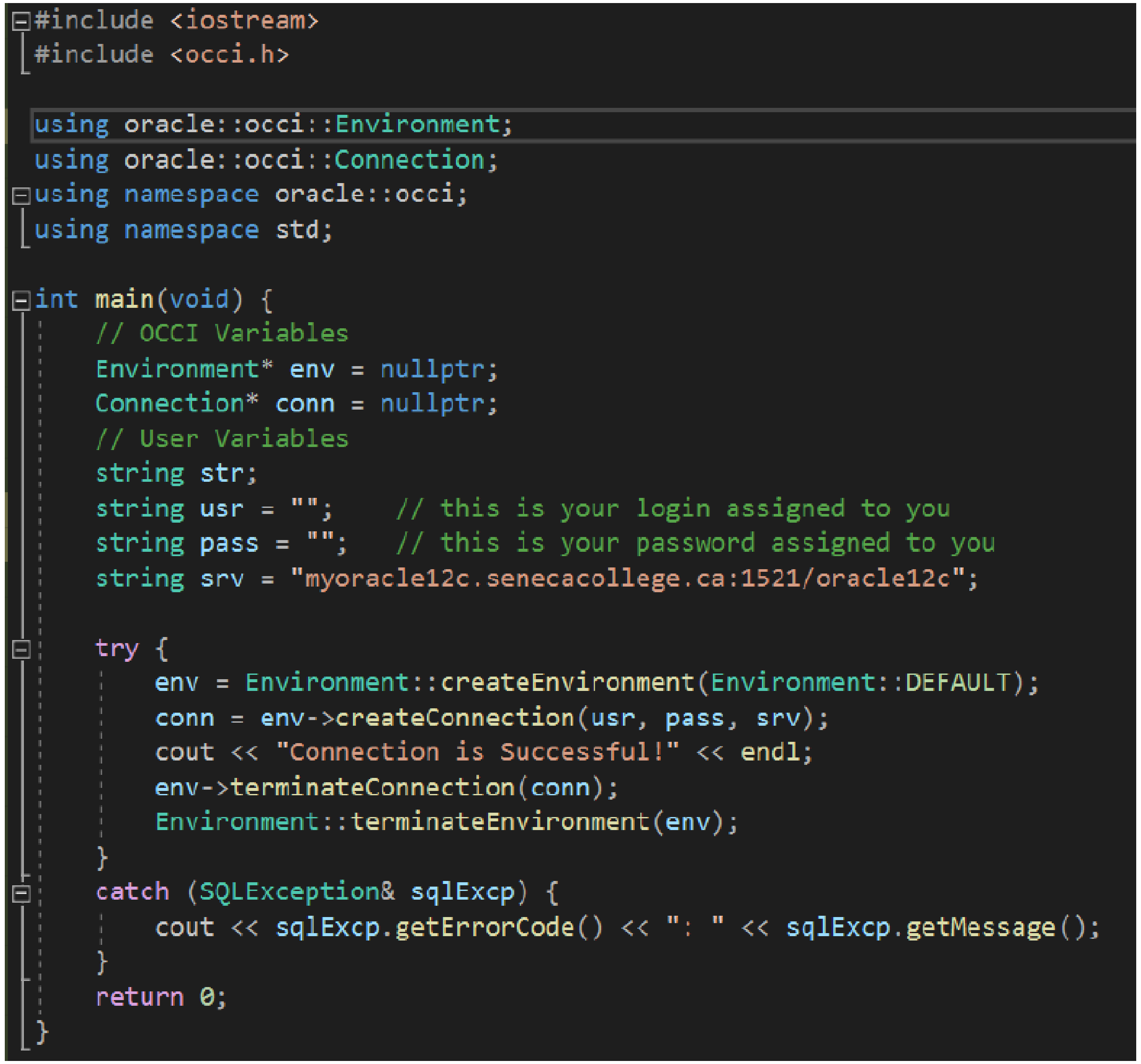
C++ User Input

```
int x;
cout << "Type a number: "; // Type a number and press enter
cin >> x; // Get user input from the keyboard
cout << "Your number is: " << x; // Display the input value
```

```
ResultSet* rs=stmt->executeQuery("SELECT
officecode,city,state,country,postalcode FROM offices ORDER BY
officecode");
while(rs->next()){

int count=rs->getInt(1);
string city=rs->getString(2);
string state=rs->getString(3);
string country=rs->getString(4);
string pc=rs->getString(5);
cout<<count<<" "<<city<<" "<<state<<" "<<country<<" "<<pc<<endl;
}
```

Example in C++ to connect to Oracle

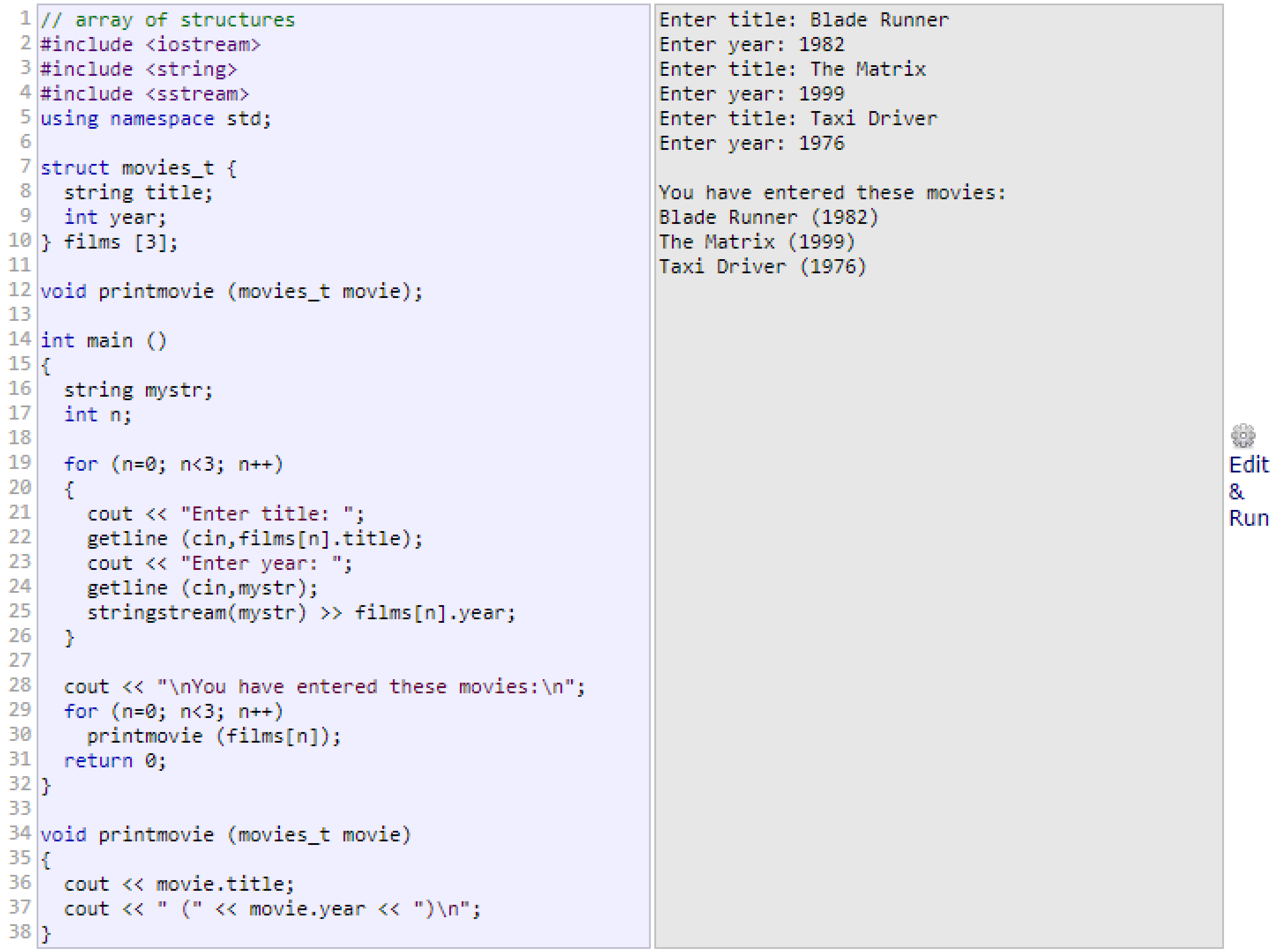


```cpp
#include <iostream>
#include <occi.h>

using oracle::occi::Environment;
using oracle::occi::Connection;
using namespace oracle::occi;
using namespace std;

int main(void) {
    // OCCI Variables
    Environment* env = nullptr;
    Connection* conn = nullptr;
    // User Variables
    string str;
    string usr = "";    // this is your login assigned to you
    string pass = "";   // this is your password assigned to you
    string srv = "myoracle12c.senecacollege.ca:1521/oracle12c";

    try {
        env = Environment::createEnvironment(Environment::DEFAULT);
        conn = env->createConnection(usr, pass, srv);
        cout << "Connection is Successful!" << endl;
        env->terminateConnection(conn);
        Environment::terminateEnvironment(env);
    }
    catch (SQLException& sqlExcp) {
        cout << sqlExcp.getErrorCode() << ": " << sqlExcp.getMessage();
    }
    return 0;
}
```



```cpp
// array of structures
#include <iostream>
#include <string>
#include <sstream>
using namespace std;

struct movies_t {
  string title;
  int year;
} films [3];

void printmovie (movies_t movie);

int main ()
{
  string mystr;
  int n;

  for (n=0; n<3; n++)
  {
    cout << "Enter title: ";
    getline (cin,films[n].title);
    cout << "Enter year: ";
    getline (cin,mystr);
    stringstream(mystr) >> films[n].year;
  }

  cout << "\nYou have entered these movies:\n";
  for (n=0; n<3; n++)
    printmovie (films[n]);
  return 0;
}

void printmovie (movies_t movie)
{
  cout << movie.title;
  cout << " (" << movie.year << ")\n";
}
```

```
Enter title: Blade Runner
Enter year: 1982
Enter title: The Matrix
Enter year: 1999
Enter title: Taxi Driver
Enter year: 1976

You have entered these movies:
Blade Runner (1982)
The Matrix (1999)
Taxi Driver (1976)
```

Some reading about databases and c++

C++ Language - C++ Tutorials - Cplusplus.com
https://www.cplusplus.com/doc/tutorial/

C++ Tutorial - W3Schools
https://www.w3schools.com/CPP/default.asp

Introduction to OCCI
https://docs.oracle.com/cd/B12037_01/appdev.101/b10778/introduction.htm

Accessing Oracle Database Using C++

https://docs.oracle.com/database/121/LNCPP/relational.htm#LNCPP003