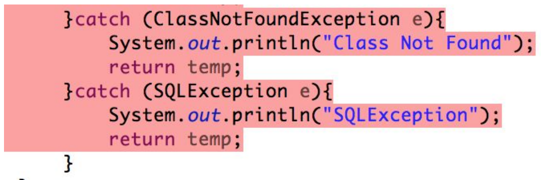
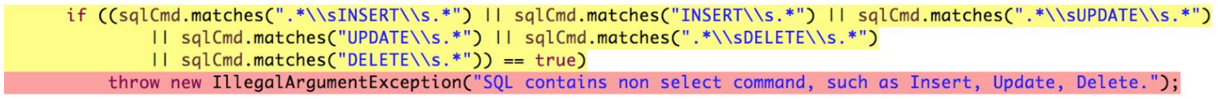
Task 3

After we carefully set the test cases, the code coverage is about 80% for the whole project. And for the level ISCRS class, statement coverage(line coverage) is to 100%(missed 0/262). The other main functional classes, such as Course, Student, StudentAndCourse etc., the misses mainly happen because the exceptions will never be raised.

1. As said above, given the definition of given interface and our implementation, the exceptions IllegalArgumentException, ClassNotFoundException, SQLException, ParseException should never be raised since we can ensure that our arguments passed to the functions in DBcoordinator will always construct valid SQL commands; all class types passed in should exist, and all SQL statements can be successfully parsed due to our upper level function definitions. If the exception is raised, it indicates there are bugs in our code such that the incorrect sql cmd is sent to the DBcoordinator.

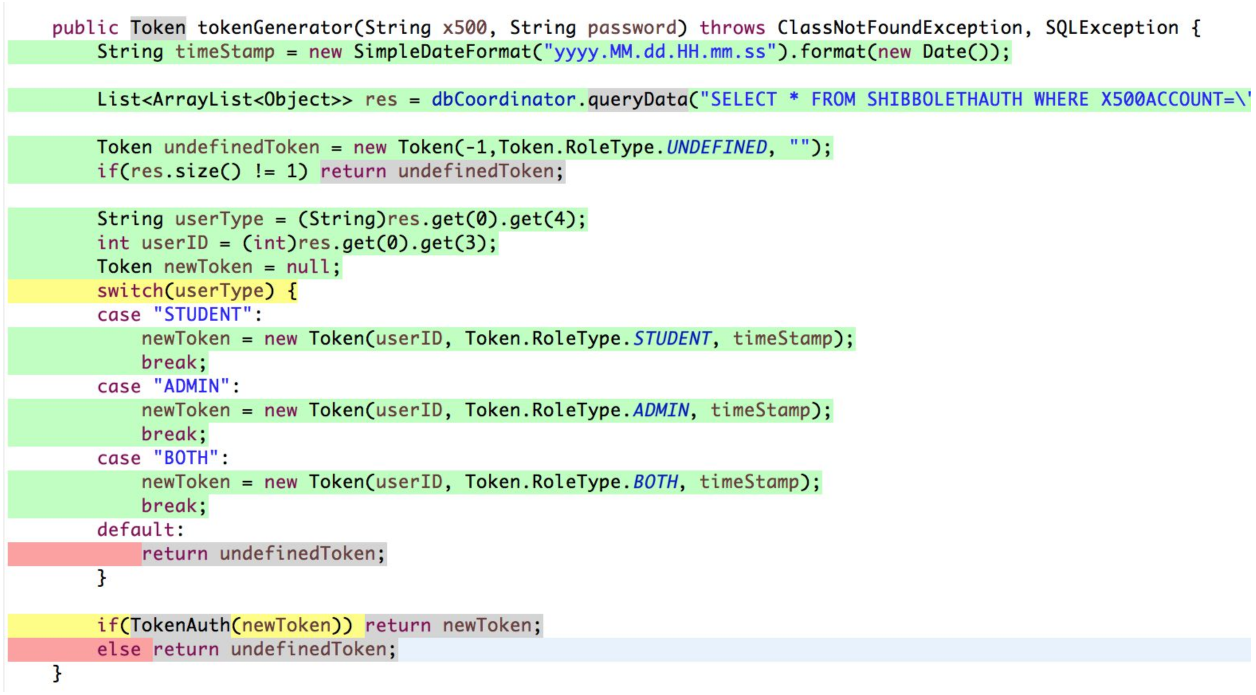


```
    }catch (ClassNotFoundException e){
        System.out.println("Class Not Found");
        return temp;
    }catch (SQLException e){
        System.out.println("SQLException");
        return temp;
    }
```



```
    if ((sqlCmd.matches(".*\\sINSERT\\s.*") || sqlCmd.matches("INSERT\\s.*") || sqlCmd.matches(".*\\sUPDATE\\s.*")
            || sqlCmd.matches("UPDATE\\s.*") || sqlCmd.matches(".*\\sDELETE\\s.*")
            || sqlCmd.matches("DELETE\\s.*")) == true)
        throw new IllegalArgumentException("SQL contains non select command, such as Insert, Update, Delete.");
```

2. Sometimes the it is the code logic that makes it impossible to cover all statements. There would be some statements that would never been executed given any input, if the code is not written well. For example, In tokenGenerator() method of ShibbolethAuth Class, if all information in the database are correct(e.g. In the ShibbolethAuth Table, there would not be some user with USERTYPE='HAHA' or anything not 'STUDENT' or 'ADMIN' or 'BOTH'), our program will never get into default case in the switch() statement as below.



```java
    public Token tokenGenerator(String x500, String password) throws ClassNotFoundException, SQLException {
        String timeStamp = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new Date());

        List<ArrayList<Object>> res = dbCoordinator.queryData("SELECT * FROM SHIBBOLETHAUTH WHERE X500ACCOUNT=\'

        Token undefinedToken = new Token(-1,Token.RoleType.UNDEFINED, "");
        if(res.size() != 1) return undefinedToken;

        String userType = (String)res.get(0).get(4);
        int userID = (int)res.get(0).get(3);
        Token newToken = null;
        switch(userType) {
        case "STUDENT":
            newToken = new Token(userID, Token.RoleType.STUDENT, timeStamp);
            break;
        case "ADMIN":
            newToken = new Token(userID, Token.RoleType.ADMIN, timeStamp);
            break;
        case "BOTH":
            newToken = new Token(userID, Token.RoleType.BOTH, timeStamp);
            break;
        default:
            return undefinedToken;
        }

        if(TokenAuth(newToken)) return newToken;
        else return undefinedToken;
    }
```

3.The lacking of data in the database also makes it difficult for us to cover all statements. Such as in the adminAddStudentToClass() and StudentAddClass() function, we have to insert some data in the database, such that there is a student who will exceed the credit limit after he/she register another course, in order to reach full line coverages.