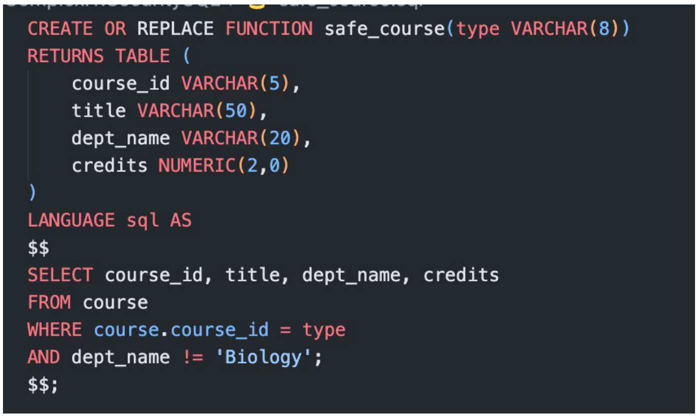
Assignment 5 – Security

Group **cit11**:

- Ida Hay Jørgensen (stud-ijoergense@ruc.dk)
- Julius Krüger Madsen (stud-juliusm@ruc.dk)
- Marek Laslo (stud-laslo@ruc.dk)
- Sofus Hilfling Nielsen (stud-sofusn@ruc.dk)

Question 1

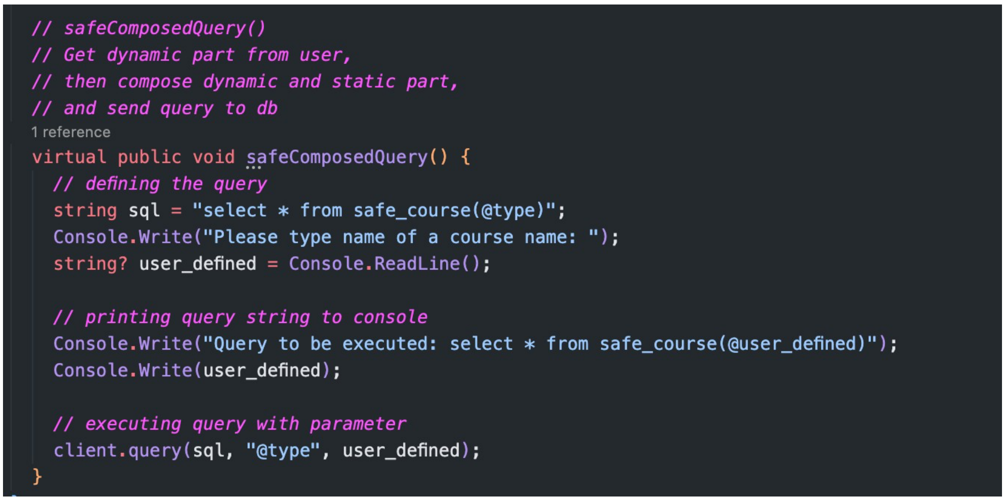
SQL Function for safe course:



```
CREATE OR REPLACE FUNCTION safe_course(type VARCHAR(8))
RETURNS TABLE (
    course_id VARCHAR(5),
    title VARCHAR(50),
    dept_name VARCHAR(20),
    credits NUMERIC(2,0)
)
LANGUAGE sql AS
$$
SELECT course_id, title, dept_name, credits
FROM course
WHERE course.course_id = type
AND dept_name != 'Biology';
$$;
```

Question 2

Method for safe composed queries:



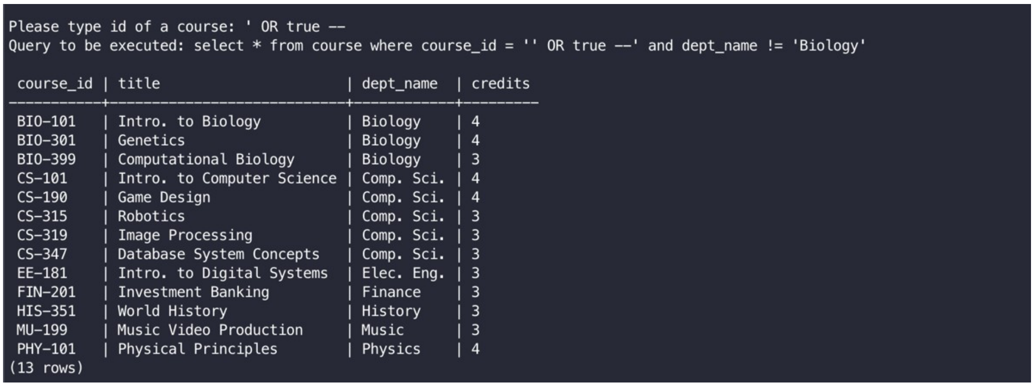
```
// safeComposedQuery()
// Get dynamic part from user,
// then compose dynamic and static part,
// and send query to db
1 reference
virtual public void safeComposedQuery() {
  // defining the query
  string sql = "select * from safe_course(@type)";
  Console.Write("Please type name of a course name: ");
  string? user_defined = Console.ReadLine();

  // printing query string to console
  Console.Write("Query to be executed: select * from safe_course(@user_defined)");
  Console.Write(user_defined);

  // executing query with parameter
  client.query(sql, "@type", user_defined);
}
```

Question 3

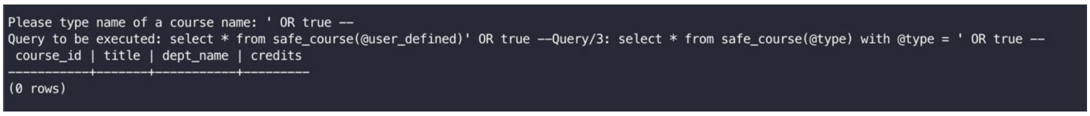
The composed query – with ' OR true --



```
Please type id of a course: ' OR true --
Query to be executed: select * from course where course_id = '' OR true --' and dept_name != 'Biology'

 course_id | title                      | dept_name  | credits
-----------+----------------------------+------------+---------
 BIO-101   | Intro. to Biology          | Biology    | 4
 BIO-301   | Genetics                   | Biology    | 4
 BIO-399   | Computational Biology      | Biology    | 3
 CS-101    | Intro. to Computer Science | Comp. Sci. | 4
 CS-190    | Game Design                | Comp. Sci. | 4
 CS-315    | Robotics                   | Comp. Sci. | 3
 CS-319    | Image Processing           | Comp. Sci. | 3
 CS-347    | Database System Concepts   | Comp. Sci. | 3
 EE-181    | Intro. to Digital Systems  | Elec. Eng. | 3
 FIN-201   | Investment Banking         | Finance    | 3
 HIS-351   | World History              | History    | 3
 MU-199    | Music Video Production     | Music      | 3
 PHY-101   | Physical Principles        | Physics    | 4
(13 rows)
```

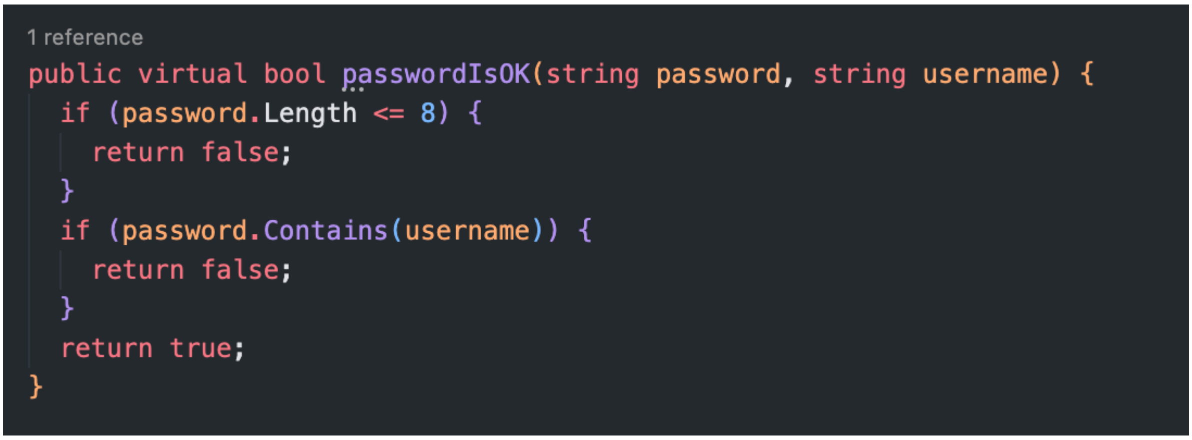
The safe composed query – with ' OR true --



```
Please type name of a course name: ' OR true --
Query to be executed: select * from safe_course(@user_defined)' OR true --Query/3: select * from safe_course(@type) with @type = ' OR true --
 course_id | title | dept_name | credits
-----------+-------+-----------+---------
(0 rows)
```

Question 4

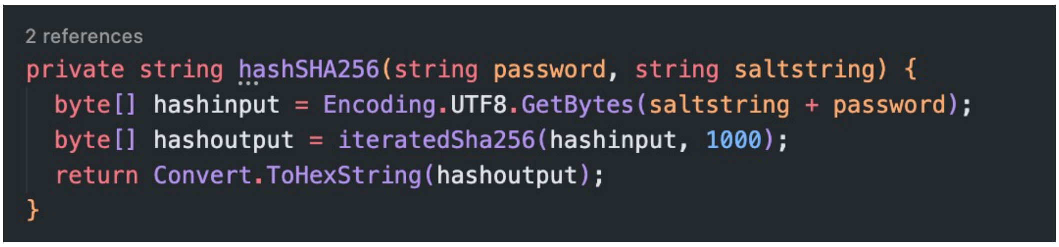
Check for password length to be over eight characters and the username is not in the password



```
1 reference
public virtual bool passwordIsOK(string password, string username) {
    if (password.Length <= 8) {
        return false;
    }
    if (password.Contains(username)) {
        return false;
    }
    return true;
}
```

Question 5

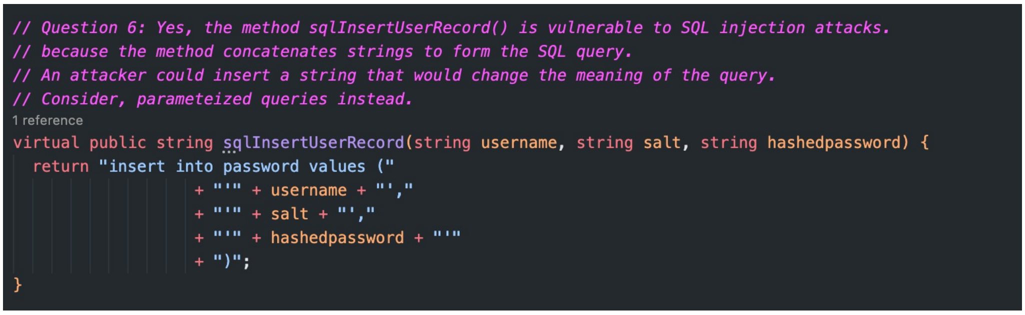
Number of iterations, 1000



```
2 references
private string hashSHA256(string password, string saltstring) {
    byte[] hashinput = Encoding.UTF8.GetBytes(saltstring + password);
    byte[] hashoutput = iteratedSha256(hashinput, 1000);
    return Convert.ToHexString(hashoutput);
}
```

Question 6

Yes, it is vulnerable to SQL injection.



```
// Question 6: Yes, the method sqlInsertUserRecord() is vulnerable to SQL injection attacks.
// because the method concatenates strings to form the SQL query.
// An attacker could insert a string that would change the meaning of the query.
// Consider, parameteized queries instead.
1 reference
virtual public string sqlInsertUserRecord(string username, string salt, string hashedpassword) {
  return "insert into password values ("
                   + "'" + username + "',"
                   + "'" + salt + "',"
                   + "'" + hashedpassword + "'"
                   + ")";
}
```