



Department of Cyber Security and Forensic Computing  
College of Computer and Cyber Sciences

## Group Project Specification

(FC332: Secure Software Design)

**Semester 2 (Spring 2023)**

### Forming Groups – [1 Mark]

- Each group will have four members.
- **Each member of a group formed with an exception and after the deadline (14<sup>th</sup> March 2023) will lose this mark.**
- Students need to form their groups and inform the instructor(s) for approval.
  - The groups formed by the students (members are chosen by the students themselves) are generally approved. However, the instructors may have to make changes to some groups.

### What is required:

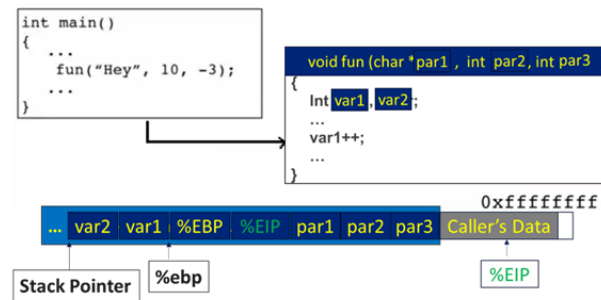
1. Demo of your application(s)
2. A report explaining your work (including Requirement Analysis, Threat Modelling, etc. where applicable)
3. Oral presentation & discussion

<b>Submission Date:</b>	<b>Assignment - 1: 13 April</b> <b>Assignment - 2: 4 May</b>
<b>Submission Form:</b>	1. Fully functional code 2. A written report (electronic, not printed) to be submitted/uploaded on <b>Teams</b>
<b>Work Type:</b>	Group Project (ideally 4 students per group).

<b>Targeted Learning Outcomes:</b>	<ul style="list-style-type: none"> <li>2.1 Develop secure and resilient software via the identification and documentation of software vulnerabilities, security requirements, and appropriate defence mechanisms. S1 (PLO 5)</li> <li>3.1 Demonstrate high competence in software security by communicating (both orally and in writing) the security threats and appropriate solutions effectively. C1 (PLO 10)</li> </ul>
<b>Marks:</b>	<b>Assignment - 1: 10 Marks</b> <b>Assignment - 2: 14 Marks</b>
<b>Warnings:</b>	<ul style="list-style-type: none"> <li>Do not use other's reports; not even as "templates".</li> <li>In case of plagiarism, you will receive a penalty.</li> </ul>
<b>Penalties:</b>	<ul style="list-style-type: none"> <li>Late submission: 1 (one) mark will be deducted for each day and 2 (two) marks will be deducted for each day from the 8th day after the due date of submission.</li> <li>Plagiarism: In case of plagiarism, your report with 50% or more plagiarism will yield maximum half of the total marks in the project (that is, your project will be evaluated out of 10 (instead of 20).</li> <li> <ul style="list-style-type: none"> <li><math>\geq 30\%</math> → You will lose 2 Marks</li> <li><math>\geq 40\%</math> → You will lose 5 Marks</li> <li><math>\geq 50\%</math> → You will lose 10 of the total marks on the project.</li> </ul> </li> <li>Changing Project Topics in less than 10 days from the date of submission will result in losing marks. The marks that you will lose will be worked out using the formula: <math>\{M/2 - M/20 \times N\}</math> where <ul style="list-style-type: none"> <li>M = Total_Marks_in_the_Project</li> <li>N = No_of_Days_from_the_Date_of_Submission</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li></li> </ul>

## First assignment:

### Stack Frame for Function Fun



<b>DESCRIPTION</b>	<p>During the lectures, your instructor has shown how parameters, variables and other elements are stored in a stack frame using PowerPoint animation.</p> <p>You will write a simple similar and <b>fully functional application in C/C++ that uses minimum two variables and takes minimum two parameters</b> as inputs .</p> <p>Use tools such as a debugger to show the live demonstration of the stack frame of a function, its contents and other relevant elements.</p>
<b>Assessment Components and Marks Distribution:</b>	<ol style="list-style-type: none"> <li>1. A Complete and fully functional application <a href="#">[1mark]</a></li> <li>2. Implement Eight of the following Nine Tasks <a href="#">[8 marks]</a> <ol style="list-style-type: none"> <li>a. Show the Variables in the stack.</li> <li>b. Show the parameters in the stock.</li> <li>c. Show the value of old EIP (return address) stored in the stack frame.</li> <li>d. Show the value of old EBP stored in the stack frame, (i.e., EBP of the caller function).</li> <li>e. Address stored in the EIP register.</li> <li>f. Address stored in the EBP register.</li> <li>g. Address of the stack pointer ESP.</li> <li>h. Address stored in the instruction pointer (address of the code that will be executed next).</li> <li>i. Content of the instruction registers (the code that is currently being executed).</li> </ol> </li> <li>3. Professionally written and well-presented Report + Live Demo. <a href="#">[1mark]</a></li> </ol>

❖ **SUGGESTED IDEAS FOR YOUR APPLICATION:**

1. Write a program in C/C++ that works as a calculator.
2. Write a program in C/C++ that reads in a sequence of characters and prints them in reverse order.
3. Write a program in C/C++ to merge two arrays of the same size sorted in descending order.
4. Write a C/C++ program to enter numbers from 1 to 7 and return the corresponding day.
5. Write a C/C++ program to find the minimum element in a stack.
6. Write a C/C++ program to calculate the average value of the stack elements.
7. Write a C/C++ program that asks you to enter three names first name second name and third name then it will put the names in any specific order (i.e., surname, first, then middle name)
8. Write a C/C++ program to enter any number and calculate its square root.
9. Write a C/C++ program to enter marks of five subjects and calculate total, average and percentage.
10. Write a C/C++ program to convert a string to lower case
11. Write a C/C++ Program to Count number of Lowercase and Uppercase Letters
12. Write a C/C++ Program to find the largest number among three number.
13. Write a C/C++ program to print day name of week.
14. Write a C/C++ program to enter month number and print number of days in month.
15. Write C/C++ program to print multiplication table of a given number
16. Write a C/C++ program to print all natural numbers in reverse order.
17. Write C/C++ program to find sum of all elements of an array
18. Write C/C++ program to count even and odd elements in an array
19. Write C/C++ program to find maximum and minimum element in array.
20. Write a C/C++ program to print all negative elements in an array
21. Write C/C++ program to count total number of negative elements in array
22. Write a C/C++ Program to Calculate Difference Between Two Time Periods
23. Write a C/C++ Program to convert Fahrenheit to Celsius
24. Write a C/C++ Program to find factorial of any number
25. Write a C/C++ Program to print number in words
26. Write a C/C++ Program to order stack element in ascending order.
27. Write a C/C++ Program to Display Fibonacci Sequence

## Second assignment:

### ❖ DESCRIPTION:

Write an application for the electronic portal of the Hajj that can be used by the Pilgrims to apply for Al-Hajj. So, the application will allow the Pilgrims to enter the following information that will be stored in a database.

- Pilgrims ID/ Iqama Number
- First Name (s) and Surname
- Date of birth
- Contact Number
- Companions
- Package company
  - Approved / Rejected
  - Comment

There will be a login page and the Pilgrims will be able to submit their applications after logging in. They should be able to modify or update their application later.

There should be a drop-down list for the Pilgrims to select their Hajj Package (Package company).

The Package company will have the ability to approve or reject an application using a drop-down list.

The Package company should be able to query the system to retrieve the records of Pilgrims so that they can approve or reject their applications and write comments, if necessary.

**Think Act and Reason:** *Does the data that comes from a drop-down list also need to be validated? Give a reason for your answer.*

<b>Second Assessment Components:</b>	<ol style="list-style-type: none"> <li>1. A Complete and fully functional application [2 mark]</li> <li>2. Security Requirement Analysis [1 mark]</li> <li>3. Threat Modelling (Identify and mitigate possible threats) [2 mark]</li> <li>4. Perform Data Validation <ol style="list-style-type: none"> <li>a. with Regular Expression [1 marks]</li> <li>b. without Regular Expression [1 mark]</li> </ol> </li> <li>5. Write queries that suffer from SQL injection. [1 mark]</li> <li>6. Use Prepared Statements (Parameterised Query). [1 marks]</li> <li>7. Use of Store Procedure [1 mark]</li> <li>8. Use Views Instead of Tables [1 mark]</li> <li>9. A well written, nicely formatted and well-presented report and presentation [1 mark]</li> <li>10. Produce a test result by evaluating the security of your application. Use at least 2 tools and techniques to perform the testing (e.g. fuzzing, SQL injection). If more than 2 is used then the additional ones <b><u>may be</u></b> considered as bonus points. [1 mark]</li> <li>11. You will apply at least <b>any three</b> of the following Code Protection Tools/Techniques: [1 mark] <ol style="list-style-type: none"> <li>a. Code Obfuscation</li> <li>b. Anti-Debugging Techniques</li> <li>c. Anti-Tamper Techniques</li> <li>d. Memory Scrambling.</li> <li>e. Code Encryption</li> <li>f. Code Virtualization:</li> </ol> </li> </ol>
--------------------------------------	--

For any further questions, please contact:

- Dr Syed Sadiqur Rahman
- Mr Mohammed Al-Khatib
- Ms Bayan Alamry