

Assignment Cover Sheet

Qualification		Module Number and Title
HD in Computing and Software Engineering /Network Technology and Cyber Security		CSE 4002 Fundamentals in Programming
Student Name & No.		Assessor
		Mrs. Nisansala Athapaththu
Hand over date		Submission Date
Assessment type	Duration/Length of Assessment Type	Weighting of Assessment
Coursework	3000 words	100%

Learner declaration
I,<name of the student and registration number>, certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Marks Awarded			
First assessor			
IV marks			
Agreed grade			
Signature of the assessor		Date	

FEEDBACK FORM
INTERNATIONAL COLLEGE OF BUSINESS & TECHNOLOGY

Module: Fundamentals in Programming

Student ID:

Assignment: Writ1

Marks Allocated	Task 1	Task 2 (a)	Task 2 (b)	Task 2 (c)	Task 2 (d)	Task 2 (e)	Task 3	Total marks
	25	15	15	10	5	5	25	100
Marks Awarded								

Task	Feedback	
1		
2		
3		
General Comment		
Assessor Name	Signature	Date

Course Work and Practical Assessment

This assignment is worth 100% of the overall assessment for this module.

Learning outcomes covered

- LO1. Explain structured programming concepts
- LO2. Design a basic structured computer program
- LO3. Developed a modularized computer programme for a prepared design
- LO4. Compile software testing and documentation

Scenario

“Upcountry Warriors” is one of the very famous baseball clubs in the city. At present the club is growing very fast but maintains player details manually. They need a computerized approach to automate player information. Every player is assigned a player registration number and a player can select a maximum of 2 teams. New players are registered by the system and the system should take information such as player registration number, first name, last name, Date of birth, runs scored, etc...

You are required to develop a C++ Application that should store the information such as players, teams, etc... The program should have a user interface which facilitates following requirements:

Main requirements are.

- Display players information
- Add new players
- Manage teams
- Search players

(More functionality can be included)

Other requirements are.

- User login.
- Logout.
- Exit.
- View Team Details.

Carefully investigate the given scenario and provide the proposed solution.

Attach softcopy of error free program with your documentation.

Keep all the backups

Viva Evaluation TASKS

Task 01. Create system requirements specification and system design diagrams (flow charts) for core functions such as Display, Add and Search player Details according to given scenario. Use appropriate modularization to reduce the complexity of the design. *(25 marks)* **(LO2)**

Task 02. Implement and submit a functional C++ program to meet the requirements given in the specification, by following the design created above. *(50 marks)* **(LO3)**

- a) Evaluate the learner's ability to describe controlling structures used for the implementation with improved coding efficiency (i.e., sequence structure, selection structure and repetition structure). *(15 marks)* **(LO1)**
- b) Identify the use of modularization with effective data passing between developed modules during the implementation. *(15 marks)* **(LO1)**
- c) Evaluate the use file handling techniques used for storage and backup requirements and use of appropriate arrays, structs(records) used. *(10 marks)* **(LO3)**
- d) Provide appropriate guidelines to user, apply validations for user inputs and improve user-friendliness of the software. *(5 marks)* **(LO3)**
- e) Assess the ease of navigation between modules, accuracy, creativity and completeness of the system. *(5 marks)* **(LO3)**

Task 03. Prepare a test document including test plan, test cases and test results. Conduct user acceptance testing and provide feedback with sample questionnaires used. *(25 marks)* **(LO4)**

Assessment Criteria

Task 01- System Design (LO2)

This submission will be assessed as follows Criteria	Total marks Allocated	Marks obtained by the student for the answer provided
	Out of 25	
Excellent Design <ul style="list-style-type: none">• Excellent SRS given in detail• Highly detailed diagram• Use of modularization concepts clearly visible• Excellent use of symbols• Clarity and Reduce complexity of the design• Backed by relevant assumptions	19-25	
Good Design <ul style="list-style-type: none">• Detail SRS including functional and non-functional requirements, data and file structure requirements• Flow charts following standard notations in flow charting and pseudo codes using proper structured English• Accurate use of selection repetition structures• Logical and continuous flow of instructions along the design	15-18	
Satisfactory Design <ul style="list-style-type: none">• Basic SRS including functional requirements• Clear identification and application of symbols in flow charts• Average level design diagrams given	11-14	
Poor Design <ul style="list-style-type: none">• Evidence of lack of understanding systems requirement specification• Poor use of design tools and symbols• Design diagrams with invalid flows, incomplete diagrams with logical errors	0-10	

Task 02 – System Implementation (LO3)

This submission will be assessed as follows Criteria	Total marks Allocated	Marks obtained by the student for the answer provided
	Out of 50	
Excellent implementation <ul style="list-style-type: none">• Excellent use of control structures with improved coding efficiency• Use file handling techniques for storage and backup requirements• Excellent Modularization with effective data passing between developed modules.• appropriate guidelines given to user, user input validation and user-friendliness of software• Easy navigation between modules, accuracy, creativity and completeness of the system	36-50	
Good implementation <ul style="list-style-type: none">• Use of Comments to improve code readability• good use of control structures with proper understanding• Modularize according to the given design.• Use of input validations, onscreen help options and User friendliness of the system	29-35	
Satisfactory implementation <ul style="list-style-type: none">• Operational system according to the requirements of the scenario• Average use of data types and operators• Average use of control structures (selection and repetition)	21-28	
Poor implementation <ul style="list-style-type: none">• Poor implementation with syntax errors• Lack of knowledge of the language basics used• Cannot fulfill basic system requirements	0-20	

Task 03 - System Testing (LO4)

This submission will be assessed as follows Criteria	Total marks Allocated	Marks obtained by the student for the answer provided
	Out of 25	
Excellent Documentation <ul style="list-style-type: none">• Excellent Test documentation with detail test plan and test cases• Acceptance test with proper questionnaire samples. Well analyzed user feedback which supports recommendations.• Testing conclusion with critical review and future recommendations• Appropriate use of language and Standard report format followed• Proper use of Referencing	19-25	
Good Documentation <ul style="list-style-type: none">• Detailed Test Plan• Appropriate Test Cases• Acceptance test with User feedback and test conclusion• Good documentation	15-18	
Satisfactory Documentation <ul style="list-style-type: none">• Basic Test Plan• Average Test Cases• Average documentation	11-14	
Poor Documentation <ul style="list-style-type: none">• Lack of test plan, poor test cases• No proper evidence of testing• Poor report formatting	0-10	
Total Marks	100	

This submission will be assessed as follows		Total marks Allocated	Marks obtained by the student for the answer provided
TASK 1		25	
TASK 2	a	15	
	b	15	
	c	10	
	d	5	
	e	5	
TASK 3		25	
TOTAL		100	

Submission Guidelines

- Submission format Report
- Paper Size: A4
- Words: 3000 words
- Printing Margins: LHS; RHS: 1 Inch
- Header and Footer: 1 Inch
- Basic Font Size: 12
- Line Spacing: 1.5
- Font Style: Times New Roman
- **Referencing should be done strictly using Harvard system**

Final Grading criteria :

Marks	Final Grade
≥ 70	Distinction
69-55	Merit
54-40	Pass
< 40	Fail