

Grade 12 Computer Science
Project Work: Part I
Full Marks: 36
Number of Questions: 6

Chapters Covered

1. Data Communication and Networking
2. Web Technology
 - o HTML
 - o CSS
 - o JavaScript
3. Programming in C
 - o Functions
 - o Pointers
 - o Structure and Union
 - o File Handling

General Instructions

1. Attempt **all six questions**.
 2. This project work should be completed **individually**.
 3. Students are expected to **research the required concepts before developing their solutions**.
 4. For programming questions, students must write, execute, test, and verify their programs.
 5. Include the **source code and sample output/screenshot** for each programming task.
 6. Use meaningful variable, function, and file names.
 7. Programs should be properly formatted and commented where necessary.
 8. For web-based questions, students should demonstrate the working webpage in a web browser.
 9. Students should mention the **resources/references used for research** at the end of the project.
 10. For networking questions, students should show relevant calculations, tables, diagrams, or other supporting work.
 11. Students should not simply copy programs or explanations from the internet. They should understand and be able to explain their work.
 12. The teacher may ask students to **demonstrate their programs and explain how they work** during evaluation.
 13. Screenshots of program output or webpage results should be clear and properly labelled.
 14. The project should be presented in a **well-organized project/practical file**.
-

Project Questions

Q.1. Web Technology: HTML, CSS and JavaScript [6 Marks]

Design and develop a simple Student Marks Calculator Web Page using HTML, CSS and JavaScript.

The webpage should contain:

- A suitable heading
- Student Name input field
- Marks obtained in three subjects
- A Calculate Result button
- A suitable area to display the result

Use HTML to create the structure of the webpage and CSS to make the page attractive and well organized.

Use simple JavaScript to calculate and display:

- Total marks
- Percentage
- Pass/Fail result

The JavaScript function should run when the user clicks the Calculate Result button.

Tasks:

a. Create the webpage using appropriate HTML elements and style it using CSS.

[3 Marks]

b. Write a simple JavaScript function that calculates the total, percentage and pass/fail result when the button is clicked.

[2 Marks]

c. Test the webpage with suitable inputs and include a screenshot of the working output.

[1 Mark]

Research requirement: Find out and briefly mention the purpose of any three HTML elements and any three CSS properties used in your project.

Q.2. Data Communication and Networking [6 Marks]

a. IPv4 Class C Addressing

Research the IPv4 Class C address and answer the following:

1. What is an IPv4 address?
2. What is the default subnet mask of a Class C network?
3. For the IP address 192.168.10.25, identify the network portion and host portion of the address.
4. Find the network address, first usable host address, last usable host address, and broadcast address for the network 192.168.10.0/24.

[3 Marks]

b. OSI Reference Model

The OSI Model divides network communication into seven layers.

1. What is the OSI Model?
2. List the seven layers of the OSI Model in correct order.
3. Write one major function of each layer.

[3 Marks]

Research requirement: Students should use a suitable textbook or reliable online reference to understand the functions of the seven OSI layers and present them in a table or diagram.

Q.3. Types of Functions in C

[6 Marks]

Research the different types of **user-defined functions in C** based on arguments and return values.

a. Explain the following four types with suitable function prototypes:

1. Function with no arguments and no return value
2. Function with arguments and no return value
3. Function with no arguments and a return value
4. Function with arguments and a return value

[3 Marks]

b. Develop a C program that uses **user-defined functions** to create a simple **Student Result Processing System**.

The program should:

- Input marks of at least five subjects.
- Calculate total marks using a function.
- Calculate percentage using another function.
- Determine the division/grade using another function.
- Display the complete result.

[3 Marks]

Q.4. Pointers in C

[6 Marks]

Research the concept of **pointers, memory addresses, address-of operator (&) and dereference operator (*)** in C.

a. Explain how a pointer stores the address of another variable and illustrate the concept with a suitable diagram or example.

[2 Marks]

b. Write and execute a C program that accepts **N integers into an array and uses pointers to find the largest and smallest values**.

The program should display:

- Entered elements
- Largest element
- Smallest element

[4 Marks]

Q.5. Structure and Record Management in C

[6 Marks]

A school wants to maintain student records containing:

- Roll Number
- Name
- Class

- Address
- Date of Birth
- Marks

a. Research and explain why **structures are useful for storing records containing different types of data**. Also mention one important difference between a **structure and a union**.

[2 Marks]

b. Define a suitable Student structure and develop a C program that:

1. Accepts information of at least **5 students**.
2. Displays all student records.
3. Searches for a student using the roll number.
4. Displays the searched student's complete information.

[4 Marks]

Q.6. File Handling: Updating an Existing File

[6 Marks]

Create a text file named **student.txt** containing records of at least **5 students**.

Each record should contain:

Roll Number, Name, Class and Marks

Develop a C program to **update the marks of an existing student in the file**.

The program should:

1. Open the existing student.txt file.
2. Display the available student records.
3. Ask the user to enter the **roll number** of the student whose marks need to be updated.
4. Search for the specified student.
5. Ask for the new marks.
6. Update the student's marks.
7. Preserve all other records.
8. Display the updated records after the modification.
9. Display an appropriate message if the specified student does not exist.

[6 Marks]

Important: The program must update an **existing file**. Students should demonstrate that the original record has actually been modified rather than simply creating a new file containing different data.