Project Work for C Programming

C Programming Project Work Guidelines and Criteria

Below is a comprehensive guide for preparing your C programming project, including rules, regulations, printing instructions, and everything needed for the hard copy submission and soft copy submission.

---

1. Rules and Regulations

a. General Rules

1. **Original Work**: The project must be your original work. Plagiarism or copied code is not allowed.
2. **Topic Approval**: All the topics should be done by the students as provided by the instructor.
3. **Deadline**: Submit the project on or before the specified deadline. **(Magh 2, 2081)**
4. **Teamwork**: If working in a team, clearly state each member's contribution.

b. Programming Rules

1. **Coding Standards**: Follow proper indentation and commenting. Use meaningful variable names.
2. **Modularity**: Break the program into multiple functions/modules for better readability.
3. **Error Handling**: Ensure the program can handle invalid inputs gracefully.
4. **Optimization**: Avoid redundant code and use efficient algorithms where possible.
5. **Documentation**: Include comments explaining the purpose of the code.

---

2. What to Include in the Hard Copy

a. Project Report Structure

The hard copy must be well-organized and include the following sections:

1. **Title Page**
   - Title of the project
   - Name(s) of the student(s)
   - Roll number(s)

   - Class, Section and Shift
   - Name of the Department HOD
   - Submission date
2. **Certificate of Authenticity**
   - A declaration stating that the project is your original work, signed by you and your instructor.
3. **Acknowledgment**
   - A short paragraph thanking those who helped you during the project.
4. **Table of Contents**
   - List all sections and their page numbers.
5. **Abstract**
   - A brief summary of the project (100–150 words).
6. **Introduction**
   - Explain the problem statement, objectives, and the importance of your project.
7. **Literature Review**
   - Discuss any prior work or research related to your project.
8. **System Design**
   - Flowcharts, pseudocode, or block diagrams explaining your project's workflow.
9. **Implementation**
   - Detailed explanation of the code and its modules.
   - Include snippets of code with explanations (not the entire source code).
10. **Source Code**
    - Attach the full, properly commented source code in an appendix.
11. **Testing and Results**
    - Provide sample inputs and outputs.
    - Include screenshots of the program's execution.
12. **Conclusion**
    - Summarize the achievements and scope for future improvements.
13. **References**
    - List all books, websites, or tutorials used during the project.

---

3. Formatting and Printing Instructions

a. Formatting

1. **Page Layout**:
   - **Page Size**: A4
   - **Margins**: 1 inch on all sides
   - **Font**: Times New Roman, 12 pt
   - **Line Spacing**: 1.5
2. **Header and Footer**:
   - Include project title in the header and page numbers in the footer.
3. **Text Alignment**:

   - Use **justified** alignment for paragraphs.
4. **Headings**:
   - Main headings: Bold, 14 pt
   - Subheadings: Bold, 12 pt

b. Printing and Binding

1. **Printing**: Use high-quality black and white printing. (Color printing optional)
2. **Binding**: Spiral or softbound binding with a transparent cover page.
3. **Cover Page Color**: White

---

4. Additional Instructions

1. **Backup**: Keep a digital copy of your report and source code.
2. **Code Submission**: Submit the source code on a USB drive or via email as provided.
3. **Consistency**: Maintain uniform formatting throughout the document.
4. **Review**: Proofread the document for typos and formatting errors before printing.
5. **Presentation**: Be prepared to explain your project during the viva .

---

5. Evaluation Criteria

a. Content (50%)

- Clarity of the problem statement and objectives
- Use of programming concepts (e.g., functions, file handling, data structures)
- Code functionality and correctness

b. Documentation (20%)

- Well-organized report
- Clear and concise explanations

c. Viva (15%)

- Ability to explain the project
- Understanding of the code and concepts

d. Innovation and Creativity (15%)

- Unique features or enhancements added to the project

Here's a structured approach for your C programming project. This outline can be tailored based on your preferences and the complexity required.

---

1. Project Title (ALL 5 project should be done by group members)

- **School Management System**
- **Library Management System**
- **Bus Reservation System**
- **Telecom Billing system**
- **Tic-tac-toe Game in C**

---

2. Objectives

Clearly state what the project aims to achieve.

Example:

- To design a system for managing records using C programming.
- To implement fundamental programming concepts like file handling, functions, and data structures.

---

3. Requirements

List the hardware and software requirements.

Example:

- **Hardware**: PC with a minimum of 4GB RAM.
- **Software**: GCC compiler, Code::Blocks, or Dev-C++.

---

4. Theory

Provide a brief explanation of the concepts used in the project, such as:

- **Data Structures**: Arrays, linked lists, or structures.

- **File Handling**: Reading and writing data to/from files.
- **Functions**: Modular design using user-defined functions.
- **Structures**:
- **Dynamic Memory Allocation**

---

5. Modules of the Project

Break down the project into smaller modules.

Example for a Library Management System:

1. **Login System**: Validate user credentials.
2. **Add Books**: Enter new book details.
3. **Search Books**: Find books by title, author, or ISBN.
4. **Borrow/Return Books**: Manage book borrowing and returning.
5. **Exit**: Save data and exit.

Example of a Bus Reservation System:

1. Login System
2. Booking of tickets
3. Cancel tickets
4. Checking bus status

Example of a School Management System:

1. Add Student Details
2. Find the student by the given roll number
3. Find the student by the given first name
4. Find the students registered in a course
5. Count of Students
6. Delete a student
7. Update Student

Example of a Telecom Billing System:

1. Add new records
2. View list of records
3. Modify records
4. View payment
5. Search Records
6. Delete records

---

6. Implementation

Detail the logic and steps for each module. Use algorithm and flowcharts.

For example:

- **Add Book Module**:
  1. Accept book details from the user.
  2. Store the details in a file or an array.
  3. Display a success message.

---

7. Source Code

Write the program in C, with proper comments explaining each step.

---

8. Testing and Debugging

Test the program with different inputs and ensure all modules work as expected. Document any bugs fixed during testing.

---

9. Results

Present sample outputs demonstrating the working of the program. Include screenshots or command-line outputs.

---

10. Conclusion

Summarize the project:

- What was achieved?
- What are the potential improvements?

---

11. References

List books, websites, or tutorials that helped during development.