

**A
PROJECT REPORT
ON
“TO DO LIST”**

SUBMITTED BY:

Miss. Wagh Samruddhi Vikram (2124UCEF1056)

SUBJECT:

**PROGRAMMING AND PROBLEM SOLVING USING
C++**

Under the guidance of

Miss. ISHWARI TIRSE



Department of Computer Science and Engineering

Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY

KOPARGAON – 423603, DIST : AHMEDNAGAR

2024-2025

INDEX

SR. NO	CONTENT	PAGE NO.
1.	INTRODUCTION	3
2.	CODE	4
3.	OUTPUT	8
4.	CONCLUSION	9

INTRODUCTION

A To-Do List is a simple application that allows users to manage their daily tasks by adding, deleting, and viewing tasks. It helps users stay organized by keeping track of pending tasks. The core functionality of a To-Do List includes the ability to create tasks, mark them as complete, and delete them once they are done.

This project is developed using C++ and Object-Oriented Programming (OOP) concepts. OOP makes the program modular, reusable, and easy to maintain by dividing the functionality into classes and objects.

CODE

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

class Task {
private:
    string taskName;
    bool isCompleted;

public:
    Task(string name) {
        taskName = name;
        isCompleted = false;
    }

    void markComplete() {
        isCompleted = true;
    }

    string getTaskName() {
        return taskName;
    }

    bool getStatus() {
        return isCompleted;
    }
};
```

```

class ToDoList {
private:
    vector<Task> tasks;

public:
    void addTask(string name) {
        Task newTask(name);
        tasks.push_back(newTask);
        cout << "Task added successfully!" << endl;
    }

    void viewTasks() {
        if (tasks.empty()) {
            cout << "No tasks in the list." << endl;
            return;
        }

        for (int i = 0; i < tasks.size(); i++) {
            cout << i + 1 << ". " << tasks[i].getTaskName();
            if (tasks[i].getStatus()) {
                cout << " (Completed)";
            }
            cout << endl;
        }
    }

    void completeTask(int taskNumber) {
        if (taskNumber > 0 && taskNumber <= tasks.size()) {
            tasks[taskNumber - 1].markComplete();
            cout << "Task marked as completed!" << endl;
        } else {
            cout << "Invalid task number!" << endl;
        }
    }
}

```

```

void deleteTask(int taskNumber) {
    if (taskNumber > 0 && taskNumber <= tasks.size()) {
        tasks.erase(tasks.begin() + taskNumber - 1);
        cout << "Task deleted successfully!" << endl;
    } else {
        cout << "Invalid task number!" << endl;
    }
}
};

```

```

int main() {
    ToDoList toDoList;
    int choice;
    string taskName;
    int taskNumber;

    do {
        cout << "\nTo-Do List Menu:" << endl;
        cout << "1. Add Task" << endl;
        cout << "2. View Tasks" << endl;
        cout << "3. Complete Task" << endl;
        cout << "4. Delete Task" << endl;
        cout << "5. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
        cin.ignore();

        switch (choice) {
            case 1:
                cout << "Enter task name: ";
                getline(cin, taskName);
                toDoList.addTask(taskName);
                break;

```

case 2:

```
    toDoList.viewTasks();  
    break;
```

case 3:

```
    cout << "Enter task number to mark as complete: ";  
    cin >> taskNumber;  
    toDoList.completeTask(taskNumber);  
    break;
```

case 4:

```
    cout << "Enter task number to delete: ";  
    cin >> taskNumber;  
    toDoList.deleteTask(taskNumber);  
    break;
```

case 5:

```
    cout << "Exiting..." << endl;  
    break;
```

default:

```
    cout << "Invalid choice! Please try again." << endl;
```

```
}
```

```
} while (choice != 5);
```

```
return 0;
```

```
}
```

OUTPUT

The screenshot shows a web browser with the URL <https://www.programiz.com/cpp-programming/online-compiler/>. The page features a Google Ads banner at the top. The main content area displays a C++ code editor with a file named `main.cpp`. The code implements a To-Do List Menu with five options: Add Task, View Tasks, Complete Task, Delete Task, and Exit. The output window shows the program's execution, including the menu display and user input for task completion and deletion.

```
main.cpp
99
100
101     case 2:
102         toDoList.viewTasks();
103         break;
104
105     case 3:
106         cout << "Enter task number to mark as complete: ";
107         cin >> taskNumber;
108         toDoList.completeTask(taskNumber);
109         break;
110
111     case 4:
112         cout << "Enter task number to delete: ";
113         cin >> taskNumber;
114         toDoList.deleteTask(taskNumber);
115         break;
116
117     case 5:
118         cout << "Exiting..." << endl;
119         break;
120
121     default:
122         cout << "Invalid choice! Please try again." << endl;
123         break;
124 } while (choice != 5);
```

Output

```
/tmp/70yChjdQ7R.o
To-Do List Menu:
1. Add Task
2. View Tasks
3. Complete Task
4. Delete Task
5. Exit
Enter your choice: 1
Enter task name: c++ assignment
Task added successfully!

To-Do List Menu:
1. Add Task
2. View Tasks
3. Complete Task
4. Delete Task
5. Exit
Enter your choice:
```


CONCLUSION

The To-Do List application is a simple but effective tool to help manage daily tasks. Using OOP principles in C++, the application is modular, easy to update, and maintain. Features like task addition, completion, and deletion have been implemented to help users track their progress efficiently.

THANK YOU!!!!!!....