



INITIAL REPORT

Group members

AMAL ABDUL QADIR (o56)

SUBJECT

Object Oriented Programming

SUBMITTED TO

SAEED UR REHMAN

DEPARTMENT OF COMPUTER SCIENCE

Table of Contents

Members:	3
Introduction:	3
Functional Requirements:	3
Object-Oriented Design Used:	4
Project Structure:	4
Conclusion :	6

“DISASTER RESPONSE SIMULATOR”

Members:

- Amal Abdul Qadir (056)

Introduction:

This project simulates emergency responses to various disasters such as earthquakes, floods, and fires. The goal is to demonstrate the use of object-oriented programming (OOP) concepts including **classes, inheritance, polymorphism, abstraction, and encapsulation**.

The system allows users to:

- Create disasters with a location, severity, and time
- Run a simulation that shows emergency response for each disaster
- Use different types of response units (rescue, medical, fire)

Functional Requirements:

1. **Create and store multiple disasters dynamically**
2. **Different disaster types** with their own behavior (Earthquake, Flood, Fire)
3. **Simulate responses** with countdown and team deployment
4. **Track availability of response units**
5. **Reset and clear simulation for new runs**

Object-Oriented Design Used:

OOP Concept	How It's Used
Class	Used to define Disaster, ResponseUnit, SimulationManager, etc.
Inheritance	Earthquake, Flood, Fire inherit from Disaster
Polymorphism	simulateResponse() is overridden in each disaster type
Encapsulation	Class members are private/protected with access methods
Abstraction	Disaster is an abstract base class with virtual functions

Project Structure:

- main.cpp: Main user interface and simulation loop
- Disaster.h/cpp: Base class
- Earthquake.h/cpp, Flood.h/cpp, Fire.h/cpp: Disaster subclasses
- ResponseUnit.h/cpp: Tracks unit availability and deployment
- SimulationManager.h/cpp: Manages the simulation

OUTPUT :

```

===== Disaster Response Simulator =====

Choose Disaster Type to Simulate:
1. Earthquake
2. Flood
3. Fire
4. Run Simulation
0. Exit
Your choice: 1
Enter location: Jegawa
Enter severity (1-10): 8
Enter timestamp (e.g., 2025-05-20 14:00): 2025-05-20 14:00

Choose Disaster Type to Simulate:
1. Earthquake
2. Flood
3. Fire
4. Run Simulation
0. Exit
Your choice: 2
Enter location: Kerala
Enter severity (1-10): 6
Enter timestamp (e.g., 2025-05-20 14:00): 2025-05-20 14:00

Choose Disaster Type to Simulate:
1. Earthquake
2. Flood
3. Fire
4. Run Simulation
0. Exit
Your choice: 4

===== BEGINNING SIMULATION =====
[EARTHQUAKE ALERT]
Location: Jegawa | Severity: 8 | Time: 2025-05-20 14:00
?? Dispatching in 3... 2... 1... GO!
????? Rescue teams dispatched to earthquake site.

```

```

===== BEGINNING SIMULATION =====
[EARTHQUAKE ALERT]
Location: Jegawa | Severity: 7 | Time: 2025-05-20 14:00
?? Dispatching in 3... 2... 1... GO!
????? Rescue teams dispatched to earthquake site.
?? Medical teams en route.
? Unit "Rescue Team" deployed to handle Earthquake.
? Unit "Medical Unit" deployed to handle Earthquake.
? Unit "Fire Brigade" deployed to handle Earthquake.
-----
? Simulation complete. All units reset.

Choose Disaster Type to Simulate:
1. Earthquake
2. Flood
3. Fire
4. Run Simulation
0. Exit
Your choice: 0
Simulation ended. Stay safe!

C:\Users\amlal\source\repos\DISASTER RESPONSE SIMULATOR-\x64\Debug\DISASTER RESPONSE SIMULATOR-.exe (process 9508) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|

```

Conclusion :

This project successfully demonstrates a working OOP simulation of a real-world system. It uses all required OOP principles in a clean and modular way.