

The background features a dark blue and black abstract design. A white line graph with three data points is visible on the left side. The data points are represented by white circles with orange centers. The line connects the top-left point to the bottom-middle point, and then to the top-right point. The overall aesthetic is technical and modern.

App Installation and
Memory Utilization in
Different Phone Models

APP INSTALLATION AND MEMORY UTILIZATION IN DIFFERENT PHONE MODELS

MEMBERS OF THE GROUP :

- DEVESH RAWAT (AP22110010038)
- PRANEETH BULUSU (AP22110010049)
- P Venkata Sri Nag (AP22110010050)
- MOHAN BALU VEDANTAM (AP2210010057)

Table of Contents :-

- 1 . Introduction
- 2 . Overview of the Topic
- 3 . Code explanation of different classes
- 4 . Conclusion

INTRODUCTION TO THE TOPIC :

■ Purpose:-

- Simulating phone models in understanding memory optimization.
- Facilitates studying resource allocation in smartphones.
- Provides insights into app management on different devices.

■ Relevance:-

- Essential for developers to optimize memory in app development.
- Understanding phone models assists in creating efficient applications.
- Relevant for understanding user experience across diverse devices.

OVERVIEW OF THE TOPIC :

■ Description :

-> Simulating Phone Models in C++ involves creating a programmatic representation of various smartphone models from different manufacturers within the C++ programming language. It encompasses functionalities such as memory allocation, app installation, and display of unique data specific to each model.

■ Objectives:-

- Memory Allocation: Simulating the allocation and usage of RAM and ROM in different phone models.

- App Installation: Implementing the installation of applications and monitoring memory consumption.

- Unique Data Display: Showcasing specific information unique to each phone model.

■ Key Features:-

- Creating classes representing distinct phone models (e.g., Oppo, Samsung, Apple).

- Utilizing inheritance to capture shared functionalities among different models.

- Implementing memory allocation checks and app installation logic.

CODE EXPLANATION OF PHONE CLASS :

Class: Phone

- Properties:
 - RAMTotal: Total RAM in GB
 - ROMTotal: Total ROM in GB
 - RAMUsed: Used RAM in GB
 - ROMUsed: Used ROM in GB
 - apps: Unordered map to store installed apps and their memory usage
- Constructor:
 - Initializes RAMTotal, ROMTotal, RAMUsed, ROMUsed, and apps
- Methods:
 - allocateSystemMemory(): Allocates system memory for apps
 - installApp(): Installs apps and manages memory usage
 - displayMemoryUsage(): Displays RAM and ROM usage statistics
 - displayAppConsumption(): Shows memory usage of installed apps

CODE EXPLANATION OF DERIVED CLASS :

- Derived Classes - Oppo, Samsung, Apple
 - Properties:
 - model: Model name of the (Oppo or Samsung or Apple) Phone
 - uniqueData: Unique information for the specific model
 - Methods:
 - Constructor: Initializes (Oppo, Samsung, Apple) -specific properties
 - displayUniqueData(): Displays unique information for the (Oppo, Samsung, Apple) model

CODE EXPLANATION - Memory Allocation Logic :

- Memory Allocation Functionality:-
 - > allocateSystemMemory() method in the Phone class
 - > Responsible for managing RAM and ROM allocation
- Key Features:-
 - > Method Purpose: Allocates memory for installed apps
 - > Parameters: Takes RAM and ROM values as input for allocation
 - > Functionality: Increments ramUsed and romUsed based on allocation
- Memory Check:-
 - > Checks if sufficient memory is available before allocation
 - > Prevents exceeding the total available memory
- Importance:-
 - > Ensures optimal utilization of available resources
 - > Avoids memory overflow or inefficiencies

CODE EXPLANATION - App Installation Process :

- ❖ App Installation Functionality:-
 - installApp() method in the Phone class
 - Manages app installation and memory utilization
- ❖ Key Features:-
 - Method Purpose: - Installs apps and tracks memory usage
 - Parameters: App name, RAM percentage, ROM percentage
 - RAM and ROM Allocation: Based on the given percentages
- ❖ installation Process:-
 - Checks for valid memory percentages before installation
 - Allocates memory if available, updates memory usage
- ❖ Optimization:-
 - Efficiently allocates memory for apps while considering available resources
 - Ensures apps are installed within the allocated memory limits

CODE :-



cpp project final code.cpp

CONCLUSION :-

- In conclusion, the simulation of phone models through C++ programming provides a structured approach to understanding memory optimization, resource allocation, and app management within diverse smartphone environments. By encapsulating functionalities such as memory allocation, app installation, and unique model-specific data representation, this approach offers developers invaluable insights into creating efficient applications that perform optimally across various devices.
- The systematic handling of memory allocation within the Phone class, coupled with the app installation process, ensures efficient utilization of device resources while preventing inefficiencies or overflow scenarios. This approach not only optimizes resource usage but also fosters a comprehensive understanding of user experiences across different phone models. Overall, the simulation of phone models in C++ stands as a foundational framework for developers, enabling them to craft applications that are both resource-efficient and adaptable to the diverse specifications of modern smartphones.



THANK YOU