

APP INSTALLATION AND MEMORY UTILIZATION IN DIFFERENT PHONE MODELS

Project submitted to the
SRM University – AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree
of

Bachelor of Technology

In

Computer Science and

Engineering School of
Engineering and Sciences(SEAS)\

Submitted by

PRANEETH (AP22110010049)

DEVESH (AP22110010038)

MOHAN (AP22110010057)

SRINAG (AP22110010050)



Under the Guidance of

Mrs. Kavitha Rani Karnena

**SRM University–AP
Neerukonda, Mangalagiri, Guntur
Andhra Pradesh – 522 240
[December, 2022]**

CERTIFICATE

DATE:01-Dec-23

This is to certify that the work present in this Project entitled “APP INSTALLATION AND MEMORY UTILIZATION IN DIFFERENT PHONE MODELS” has been carried out by **DEVESH, PRANEETH, MOHAN, SRINAG** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Mrs. Kavitha Rani Karnena

Acknowledgement

We the teams members would like to express our special thanks to Mrs. Kavitha Rani Karnena who gave us the golden opportunity to do this wonderful project on the topic "APP INSTALLATION AND MEMORY UTILIZATION IN DIFFERENT PHONE MODELS", which also helped in doing a lot of Research work on the concept.

Secondly, I would also like to thank my team members who engaged in this project.

INDEX

Certificate.....	1
Acknowledgments.....	2
Introduction.....	3
Code Implementation.....	4
Sample Output.....	8
Concluding Remarks.....	10

INTRODUCTION:

Purpose:

- Simulating phone models in understanding memory optimization.
- Provides insights into app managements on different devices.
- Essential for developers to optimize memory in app management.
- Relevant for understanding user experience across diverse devices.

Code Implementation:

```
o1.cpp > ...
#include <iostream>
#include <unordered_map>
#include <ctime>
#include <algorithm>

using namespace std;

// Base class: Phone
class Phone {
protected:
    int ramTotal; // Total RAM in GB
    int romTotal; // Total ROM in GB
    int ramUsed; // Used RAM in GB
    int romUsed; // Used ROM in GB
    unordered_map<string, double> apps; // Installed apps and their memory usage

public:
    Phone(int ram, int rom) : ramTotal(ram), romTotal(rom), ramUsed(0), romUsed(0) {}

    void allocateSystemMemory(int ram, int rom) {
        ramUsed += ram;
        romUsed += rom;
    }

    void installApp(string appName, double ramPercent, double romPercent) {
        float ram = static_cast<int>((ramPercent) / 1);
        int rom = static_cast<int>((romTotal * romPercent) / 100);

        if (ramUsed + ram <= ramTotal && romUsed + rom <= romTotal) {
            apps[appName] = ram;
        }
    }
};
```

```

    string model; // Model of the Apple phone
    string uniqueData; // Unique data for the specific model

public:
    Apple(int ram, int rom, string modelName, string data) : Phone(ram, rom), model(modelName), uniqueData(data) {}

    void displayUniqueData() {
        cout << "Unique data for Apple " << model << ": " << uniqueData << endl;
    }
};

// Function to generate random model for Samsung and Apple phones
string generateRandomModel() {
    string models[] = {"A", "B", "C", "D"}; // Sample models
    int idx = rand() % 4; // Random index from 0 to 3
    return models[idx];
}

int main() {
    srand(static_cast<unsigned int>(std::time(nullptr))); // Seed for random number generation

    string companyName, modelName;
    cout << "Enter the company name: ";
    cin >> companyName;

    // Convert the input company name to lowercase for consistency
    transform(companyName.begin(), companyName.end(), companyName.begin(), ::tolower);

    if (companyName == "oppo") {
        cout << "Enter the Oppo model name (A or B): ";
    }
}

```

```

private:
    string model; // Model of the Oppo phone
    string uniqueData; // Unique data for the specific model

public:
    Oppo(int ram, int rom, string modelName, string data) : Phone(ram, rom), model(modelName), uniqueData(data) {}

    void displayUniqueData() {
        cout << "Unique data for Oppo " << model << ": " << uniqueData << endl;
    }
};

// Derived class: Samsung
class Samsung : public Phone {
private:
    string model; // Model of the Samsung phone
    string uniqueData; // Unique data for the specific model

public:
    Samsung(int ram, int rom, string modelName, string data) : Phone(ram, rom), model(modelName), uniqueData(data) {}

    void displayUniqueData() {
        cout << "Unique data for Samsung " << model << ": " << uniqueData << endl;
    }
};

// Derived class: Apple
class Apple : public Phone {
private:
    string model; // Model of the Apple phone
    string uniqueData; // Unique data for the specific model

public:
    Apple(int ram, int rom, string modelName, string data) : Phone(ram, rom), model(modelName), uniqueData(data) {}

    void displayUniqueData() {
        cout << "Unique data for Apple " << model << ": " << uniqueData << endl;
    }
};

// Function to generate random model for Samsung and Apple phones
string generateRandomModel() {
    string models[] = {"A", "B", "C", "D"}; // Sample models
    int idx = rand() % 4; // Random index from 0 to 3
    return models[idx];
}

int main() {
    srand(static_cast<unsigned int>(std::time(nullptr))); // Seed for random number generation

    string companyName, modelName;
    cout << "Enter the company name: ";
    cin >> companyName;

    // Convert the input company name to lowercase for consistency
    transform(companyName.begin(), companyName.end(), companyName.begin(), ::tolower);

    if (companyName == "oppo") {
        cout << "Enter the Oppo model name (A or B): ";
    }
}

```



```

cin >> modelName;
transform(modelName.begin(), modelName.end(), modelName.begin(), ::tolower); // Convert model name
if (modelName == "a") {
    Oppo oppoModelA(8, 128, "Model A", "Unique data for Model A");
    oppoModelA.allocateSystemMemory(2, 20);
    oppoModelA.installApp("Instagram", 0, 2);
    oppoModelA.installApp("WhatsApp", 0, 3);
    oppoModelA.displayMemoryUsage();
    oppoModelA.displayAppConsumption();
    oppoModelA.displayUniqueData();
} else if (modelName == "b") {
    Oppo oppoModelB(6, 64, "Model B", "Unique data for Model B");
    oppoModelB.allocateSystemMemory(1, 10);
    oppoModelB.installApp("PUBG", 20, 15);
    oppoModelB.installApp("WhatsApp", 4, 3);
    oppoModelB.displayMemoryUsage();
    oppoModelB.displayAppConsumption();
    oppoModelB.displayUniqueData();
} else {
    cout << "Invalid Oppo model name" << endl;
}
} else if (companyName == "samsung") {
    cout << "Enter the Samsung model name (A, B, C, D): ";
    cin >> modelName;
    transform(modelName.begin(), modelName.end(), modelName.begin(), ::tolower); // Convert model name
    Samsung samsung(12, 256, modelName, "Unique data for " + modelName);
    samsung.allocateSystemMemory(3, 30);
    samsung.installApp("WhatsApp", 5, 3);
    samsung.installApp("PUBG", 15, 7);
    samsung.displayMemoryUsage();
    samsung.installApp("WhatsApp", 5, 3);
    samsung.installApp("PUBG", 15, 7);
    samsung.displayMemoryUsage();
    samsung.displayAppConsumption();
    samsung.displayUniqueData();
} else if (companyName == "apple") {
    cout << "Enter the Apple model name (A, B, C, D): ";
    cin >> modelName;
    transform(modelName.begin(), modelName.end(), modelName.begin(), ::tolower); // Convert model name
    Apple apple(6, 128, modelName, "Unique data for " + modelName);
    apple.allocateSystemMemory(2, 16);
    apple.installApp("Instagram", 4, 2);
    apple.installApp("WhatsApp", 5, 3);
    apple.displayMemoryUsage();
    apple.displayAppConsumption();
    apple.displayUniqueData();
} else {
    cout << "Invalid company name" << endl;
}

return 0;
}

```

Sample Output:

Output for company name “Apple” with mode “A”

```
Output
/tmp/guXj3SrlcE.o
Enter the company name: apple
Enter the Apple model name (A, B, C, D): A
App 'Instagram' installed with 4GB RAM and 2GB ROM
The RAM consumption for this app is 4GB
Insufficient memory to install app 'WhatsApp'
RAM Total: 6GB
RAM Used: 6GB
RAM Free: 0GB
ROM Total: 128GB
ROM Used: 18GB
ROM Free: 110GB
App Memory Usage:
Instagram - RAM: 4GB
Unique data for Apple a: Unique data for a
```

Output for company name “Samsung” with model “B”

```
Output
/tmp/guXj3Sr1cE.o
Enter the company name: samsung
Enter the Samsung model name (A, B, C, D): B
App 'WhatsApp' installed with 5GB RAM and 7GB ROM
The RAM consumption for this app is 5GB
Insufficient memory to install app 'PUBG'
RAM Total: 12GB
RAM Used: 8GB
RAM Free: 4GB
ROM Total: 256GB
ROM Used: 37GB
ROM Free: 219GB
App Memory Usage:
WhatsApp - RAM: 5GB
Unique data for Samsung b: Unique data for b
```

Concluding Works:

The simulation of phone models through C++ programming provides a structured approach to understanding memory optimization, resource allocation, and app management within diverse smart phone 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 ensures efficient utilization of device resources while preventing inefficiencies. Overall, the simulation of phone models in C++ stands as a foundational framework for developers, enabling them to craft applications efficient that are both both resource efficient and easy to adapt them for the diverse specifications for smartphone models.