# VIT - Vellore

Name: RONIT MEXSON .
Email: ronit.mexson2024@vitstudent.ac.in
Roll no: 24BAI0036
Phone: 9999999999
Branch: ARUMUGA ARUN R_OOPS
Department: admin
Batch: VL2024250502365
Degree: admin

Scan to verify results

## BCSE102P_Structured and Object Oriented Programming Lab_VL2024250502365

### VIT V_Structured and OOP_Lab 7_COD_Medium_Pure Virtual Functions

Attempt : 1
Total Mark : 10
Marks Obtained : 10

### Section 1 : Coding

1. Problem Statement

Design a program for a transport system operation. Create a base class Transport with a pure virtual function named operate. Implement two derived classes, TransportA and TransportB, which calculate and display the time taken to move between locations based on the transport type.

The program takes input for the number of transport systems and their operations, then outputs the time taken in seconds for each operation.

For TransportA: Time Taken = abs(start location - destination) * 2

For TransportB: Time Taken = abs(start location - destination) * 3

***Answer***

```
// You are using GCC
#include <iostream>
#include <cmath>
using namespace std;

class Transport {
public:
    virtual void operate(int start, int destination) = 0;
};

class TransportA : public Transport {
public:
    void operate(int start, int destination) override {
        int time = abs(start - destination) * 2;
        cout << "Time taken: " << time << " seconds" << endl;
    }
};

class TransportB : public Transport {
public:
    void operate(int start, int destination) override {
        int time = abs(start - destination) * 3;
        cout << "Time taken: " << time << " seconds" << endl;
    }
};

int main() {
    int N;
    cin >> N;

    for (int i = 0; i < N; i++) {
        char type;
        int start, destination;
        cin >> type >> start >> destination;

        Transport* transport = nullptr;

        if (type == 'P' || type == 'p') {
            transport = new TransportA();
        } else if (type == 'F' || type == 'f') {
```

```
            transport = new TransportB();
        }

        if (transport != nullptr) {
            transport->operate(start, destination);
            delete transport;
        }
    }

    return 0;
}
```

***Status :*** Correct **Marks : 10/10**