A

PROJECT REPORT

ON

“Tax Calculation System"

SUBMITTED BY:

Mr. Tanvesh Yeole (2124UCEM1012)

SUBJECT:

CORE C++ PROGRAMMING

Under the guidance of

Miss. ISHWARI TIRSE.

Department of Computer Science and Engineering

Sanjivani Rural Education Society's

SANJIVANI UNIVERSITY

KOPARGAON-423601, DIST: AHMEDNAGAR

2024-2025

INDEX

INTRODUCTION

A tax calculation system is essential for determining the taxes owed by individuals or businesses based on their income, expenditures, and other factors. This system automates the process, ensuring accuracy and efficiency. In this program, we demonstrate a tax calculation system where users input their income, and the system calculates the tax based on different tax brackets.

CODE

```
#include <iostream>
#include <vector>
#include <string> using
namespace std; // Base
class for taxpayer class
TaxPayer { protected:
string name;     double
income;

public:
   TaxPayer(string n, double i) : name(n), income(i) {}

   virtual double calculateTax() = 0; // Pure virtual function for
calculating tax     void displayTax() {
       cout << "Tax details for " << name << ":" << endl;
cout << "Income: " << income << endl;         cout <<
"Tax Amount: " << calculateTax() << endl;
   }
};
```

```
// Derived class for individual taxpayers class
Individual : public TaxPayer { public:

   Individual(string n, double i) : TaxPayer(n, i) {}


   double calculateTax() override {
double tax = 0;         if (income <=
50000) {
          tax = income * 0.05; // 5% tax for income <= 50,000
       } else if (income <= 100000) {
          tax = 50000 * 0.05 + (income - 50000) * 0.1; // 10% tax for
income > 50,000 and <= 100,000
       } else {
          tax = 50000 * 0.05 + 50000 * 0.1 + (income - 100000) * 0.2; //
20% tax for income > 100,000
       }
       return tax;
   }
       };
// Function to simulate tax calculation void
taxCalculationSystem() {
vector<TaxPayer*> taxPayers;
```

```
    taxPayers.push_back(new Individual("John Doe", 45000));
taxPayers.push_back(new Individual("Jane Smith", 120000));
taxPayers.push_back(new Corporate("TechCorp", 200000));
taxPayers.push_back(new Corporate("MegaCorp", 750000));

    for (TaxPayer* payer : taxPayers) {         payer-
>displayTax();
        cout << "-------------------------" << endl;
    }
    // Clean up dynamic memory     for
(TaxPayer* payer : taxPayers) {
delete payer;
    }
}

int main() {
taxCalculationSystem();     return
0;
}
```

OUTPUT

Tax details for John Doe:

Income: 45000

Tax Amount: 2250

---------- Tax

details for Jane Smith:

Income: 120000

Tax Amount: 11500

---------- Tax

details for TechCorp:

Income: 200000

Tax Amount: 25000

---------- Tax

details for MegaCorp:

Income: 750000

Tax Amount: 195000

CONCLUSION

This program demonstrates a basic tax calculation system that distinguishes between individual and corporate taxpayers. By using object-oriented principles like inheritance and polymorphism, we effectively handle different tax rates based on income brackets. This system ensures accurate tax calculations and can be easily extended to include more complex tax rules or additional taxpayer categories. Such systems can save significant time and reduce errors in financial operations.

THANK YOU…