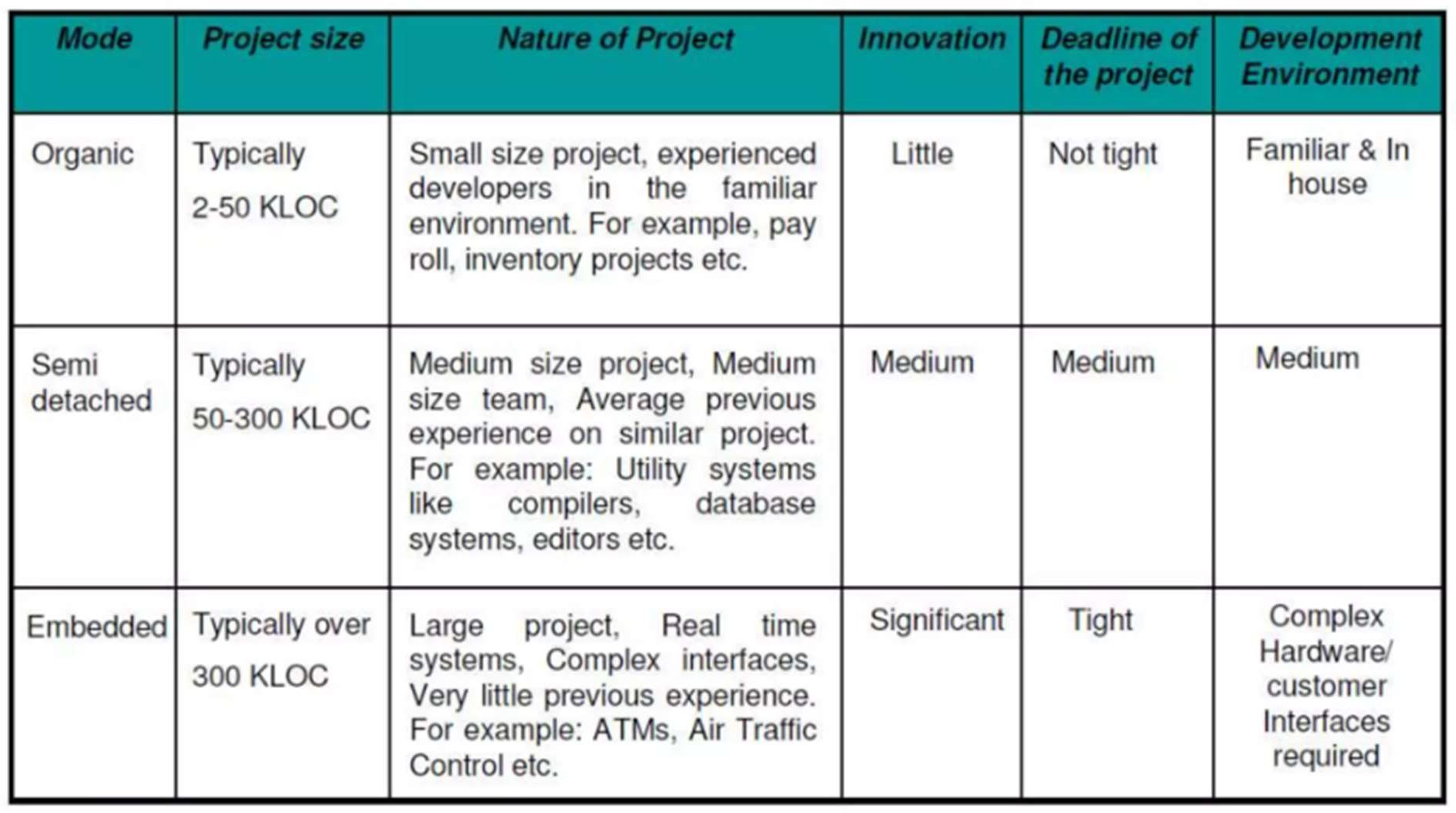
Practical 4: Cost Estimation using COCOMO Model

**THEORETICAL DISCUSSION:**

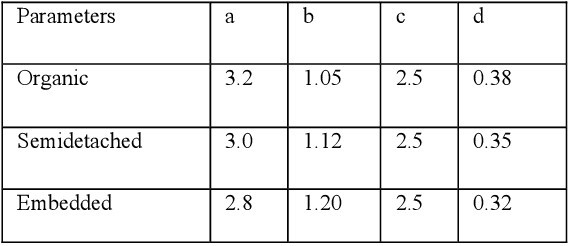
Cost Estimation using COCOMO Model

* Organic
* Semi - Detached
* Embedded



1. Basic COCOMO Model:-

E = a(KLOC)b E = effort applied in person-months D = c(E)d D = development tie in months



Average Staff Size (SS) = E/D persons Productivity (P) = (KLOC/E)KLOC/PM

# Code:-

#include <iostream> #include <cmath>

using namespace std;

// Function to calculate the effort required for a project double calculateEffort(double size, double a, double b)

{

return a \* pow(size, b);

}

// Function to calculate the development time for a project double calculateTime(double effort, double c, double d)

{

return c \* pow(effort, d);

}

// Function to calculate the number of people required for a project double calculatePeople(double effort, double time)

{

return effort / time;

}

// Function to calculate the type of project from the size of the project int getType(int size)

{

if (size >= 0 && size <= 50) return 0;

else if (size > 50 && size <= 300) return 1;

else

return 2;

}

int main()

{

double size, a, b, c, d;

double table[3][4] = {{2.4, 1.05, 2.5, 0.38},

{3.0, 1.12, 2.5, 0.35},

{3.6, 1.20, 2.5, 0.32}};

// Get input from user

cout << "Enter the size of the project (in KLOC): "; cin >> size;

int type = getType(size);

// Set values of constants a, b, c, and d a = table[type][0];

b = table[type][1]; c = table[type][2]; d = table[type][3];

// Calculate the effort, time, and number of people required for the project

double effort = calculateEffort(size, a, b); double time = calculateTime(effort, c, d); double people = calculatePeople(effort, time);

// Output the results

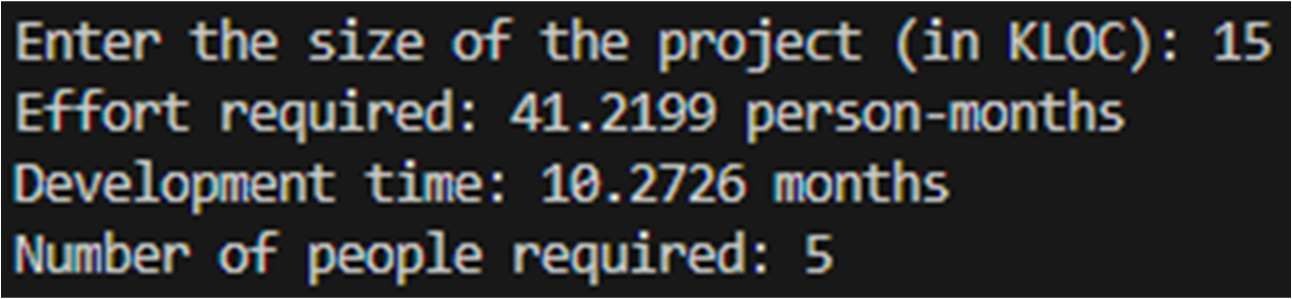
cout << "Effort required: " << effort << " person-months" << endl; cout << "Development time: " << time << " months" << endl;

cout << "Number of people required: " << ceil(people) << endl;

return 0;

}

# Output:-



1. Intermediate Model **–** The basic Cocomo model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software systems. However, in reality, no system’s effort and schedule can be solely calculated on the basis of Lines of Code. For that, various other factors such as reliability, experience, and Capability. These factors are known as Cost Drivers and the Intermediate Model utilizes 15 such drivers for cost estimation. Classification of Cost Drivers and their Attributes:

# -- Product attributes –

* Required software reliability extent
* Size of the application database
* The complexity of the product
* Run-time performance constraints
* Memory constraints
* The volatility of the virtual machine environment
* Required turnabout time
* Analyst capability
* Software engineering capability
* Applications experience
* Virtual machine experience
* Programming language experience
* Use of software tools
* Application of software engineering methods
* Required development schedule

# Code:-

#include <iostream> #include <cmath>

using namespace std;

double calculateEffort(double size, double a, double b, double c) { return a \* pow(size, b) \* pow(c, 0.35);

}

double calculateTime(double effort, double d, double e, double f) { return d \* pow(effort, e) \* pow(f, 0.35);

}

double calculatePeople(double effort, double time) { return effort / time;

}

int main() {

double size, a, b, c, d, e, f;

cout << "Enter the size of the project (in KLOC): "; cin >> size;

cout << "Enter the values of a, b, and c: "; cin >> a >> b >> c;

cout << "Enter the values of d, e, and f: "; cin >> d >> e >> f;

double effort = calculateEffort(size, a, b, c); double time = calculateTime(effort, d, e, f); double people = calculatePeople(effort, time);

cout << "Effort: " << effort << " person-months" << endl; cout << "Time: " << time << " months" << endl;

cout << "Number of people required: " << ceil(people) << endl;

return 0;

}

**Output:-**

