

Semester	T.E. Semester V – Computer Engineering
Subject	Software Engineering
Subject Professor In-charge	Dr. Sachin Bojewar
Assisting Teachers	Prof. Sneha Annappanavar
Laboratory	M313B

Student Name	Deep Salunkhe
Roll Number	21102A0014
TE Division	A

Title: To study the concept of cocomo.

Explanation:

COCOMO, which stands for CONstructive COSt MOdel, is a vital tool in the field of software engineering, specifically designed to assist in estimating the cost, effort, and time required for developing a software project. This model, first introduced by Dr. Barry Boehm in 1981, plays a crucial role in project management, enabling project managers, software developers, and stakeholders to make informed decisions about project scheduling, resource allocation, and budgeting. It offers a structured and systematic approach to software project estimation.

The COCOMO model comes in various versions, with COCOMO II being the most commonly used and comprehensive one. At its core, COCOMO employs a mathematical formula to estimate the effort (E) needed for a project. This estimation depends on the size of the software project, typically measured in thousands of lines of code (KLOC). The formula is expressed as $E = a * (KLOC)^b$. The constants 'a' and 'b' are key variables, and their values are influenced by the project's complexity, which COCOMO categorizes into three main types: organic, semi-detached, and embedded.

Organic projects, the simplest category, involve experienced teams and well-understood requirements. Semi-detached projects lie in the middle of the complexity spectrum, while embedded projects are the most intricate, characterized by tight constraints and significant uncertainties in the project's requirements.

For a more refined estimation, COCOMO II takes into account additional factors that affect software development. It goes beyond project complexity to consider elements like the development process, personnel capabilities, and project-specific attributes. COCOMO II offers various sub-models, each suited for specific project phases, such as the Application Composition Model, Early Design Model, and Post-Architecture Model. These models provide a more accurate and detailed analysis of software project costs and effort throughout the project's life cycle.

It's important to note that while COCOMO is a valuable tool for initial cost and effort estimation, real-world projects often deviate from these estimates. Thus, COCOMO should be used in conjunction with other project management techniques and adapted as the project progresses. The model is a parametric one, relying on historical data and expert judgment, and as such, it should serve as a helpful guide rather than an absolute guarantee of project success.

Implementation:

```
#include <bits/stdc++.h>

using namespace std;

int fround(float x)
{
    int a;
    x = x + 0.5;
    a = x;
    return (a);
}

void calculate(float table[][4], int n, char mode[][15], int size)
{
    float effort, time, staff;
    int model;

    if (size >= 2 && size <= 50)
        model = 0;
    else if (size > 50 && size <= 300)
        model = 1;
    else if (size > 300)
        model = 2;

    cout << "The mode is " << mode[model];

    effort = table[model][0] * pow(size, table[model][1]);
    time = table[model][2] * pow(effort, table[model][3]);
    staff = effort / time;

    cout << "\nEffort = " << effort << " Person-Month";
    cout << "\nDevelopment Time = " << time << " Months";
    cout << "\nAverage Staff Required = " << fround(staff) << " Persons";
}

int main()
{
    float 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 };
    char mode[][15] = { "Organic", "Semi-Detached", "Embedded" };
    int size = 4;

    calculate(table, 3, mode, size);

    return 0;
}
```

Output:

```
The mode is Organic
Effort = 9.6006 Person-Month
Development Time = 2.15795 Months
Average Staff Required = 4 Persons
```

Conclusion:

Offers a simplified implementation of the Basic COCOMO model for software project estimation. It classifies projects into three categories based on size (Organic, Semi-Detached, and Embedded) and calculates essential project parameters like effort, development time, and the average staff required. The program demonstrates the use of mathematical formulas and basic conditional statements to estimate project attributes. However, it's important to note that the program is a highly simplified representation of COCOMO and does not encompass the full complexity and nuance involved in real-world software project estimation. COCOMO models used in practice incorporate a wide range of factors, including personnel capability, development process, and project-specific attributes, to provide more accurate and detailed estimates for software development projects.