Q) COCOMO model.

=> (1) Constructive cost model is one of the most used software estimation model.
(2) This model depends on size i.e number of lines of code for software product development.
(3) It estimates efforts required for the project, total project cost and scheduled time.
(4) The key parameters of COCOMO are primarily effort & schedule:
(i) Effort: Amount of labor that will be required to complete a task.
(ii) Schedule: The amount of time required for completion of job.
(5) Types of project in COCOMO model:
(a) Organic project
(i) Project is small & simple.
(ii) Few requirements in project.
(iii) Product team is small
(iv) Problem is well understood & has been solved in the past.
(b) semi-detached
(i) Medium size and mixed rigid requirement
(ii) Project's complexity is not too high or not too low.
(iii) Both experienced & non-experienced members are required in the team.
(c) Embedded
(i) Large project with fixed requirements
(ii) Larger team size

(iii) Highest level of complexity, creativity and experience required.

(6) Formulas for COCOMO Model:

$E = a * (KLOC)^b$

$D = c * (E)^d$

where

a, b, c, d are constant parameters

E = Total efforts required for project

D = Total time required for project development

(7) Basic COCOMO coefficients

Software project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-dettached	3.0	1.12	2.5	0.35
embedded	3.6	1.20	2.5	0.32

Q Suppose that a project was estimate to 400 KLOC. Calculate the effort and development time for each of the three modes.

=> Basic COCOMO eqn:

$E = a \times (KLOC)^b$

$D = c \times (E)^d$

Person estimation = E/D (optional)

Productivity = KLOC / E

(1) Organic mode

$E = 2.4 \times (400)^{1.05}$

$= 1295.31$ PM

$D = 2.5 \times (1295.31)^{0.38}$

$= 38.07$ PM

) Semi - detached mode

$E = 3.0 \times (400)^{1.12}$

$= 2462.79$ PM

$D = 2.5 \times (2462.79)^{0.35}$

$= 38.45$ PM

) Embedded mode

$E = 3.6 \times (400)^{1.20}$

$= 4772.81$ PM

$D = 2.5 \times (4772.8)^{0.32}$

$= 38$ PM

Module 3

Q1. Project Scheduling and Tracking

=>

1) Project Scheduling involves planning and organizing tasks and resources to complete a project on time.
2) The project manager breaks down the project into smaller tasks and estimates the time and resources needed for each.
3) Effective scheduling helps reduce costs and improves customer satisfaction.
4) **Project Scheduling Process:**
    i) Identify all tasks and functions required.
    ii) Break down larger functions into smaller tasks.
    iii) Determine task dependencies.
    iv) Allocate resources and assign people to tasks.
    v) Plan start and end dates for each task.
    vi) Use tools like Gantt charts or Activity Networks to visualize the schedule.
5) **Project Scheduling Techniques:**
    i) Critical Path Method (CPM): Identifies the longest path of dependent tasks to estimate minimum project duration.
    ii) Program Evaluation and Review Technique (PERT): Uses optimistic, pessimistic, and most likely time estimates to calculate expected duration.
    iii) Gantt Chart: A bar chart representing tasks with start and end dates, showing task dependencies and deadlines.
6) Project Tracking is the process of monitoring progress to ensure the project stays on track with the planned schedule, budget, and quality.
7) Tracking helps detect delays and issues early, allowing the project manager to make adjustments.
8) **Project Tracking Process:**
    i) Set milestones as checkpoints.
    ii) Monitor task completion and compare with the planned schedule.
    iii) Track budget usage and make adjustments as necessary.
    iv) Use tracking tools like dashboards and progress reports.
    v) Hold regular status meetings to discuss progress and address issues.

Q2. What is cost estimation? Explain LOC method

=>

Cost Estimation

1) Software Cost Estimation is a process used to predict the approximate cost of a software project before development begins.
2) This prediction helps in budgeting and planning for resources effectively.
3) Cost estimation involves calculating the total financial resources needed to complete the project, which includes development and testing efforts in software engineering.
4) Factors considered for project budgeting:
   i) Specification & Scope
   ii) Location
   iii) Duration
   iv) Project Team Efforts
   v) Resources

LOC (Lines of Code) Method

1) LOC (Lines of Code) is a size-oriented metric that focuses on measuring the physical size of a software program by counting the lines of code written.
2) It is one of the earliest and simplest methods to estimate software size and productivity.
3) Consider a simple code example to illustrate how LOC is calculated:

```
// Import header file
#include <iostream>
int main() {
    int num = 10;  // Initialize number
    // Logic to check if number is even
    if (num % 2 == 0) {
        std::cout << "It is an even number";
    }
    return 0;
}
```

In this example:

a. Total Lines: 9
b. Blank or comment lines: 2 (first and third lines)
c. LOC: 7 (after excluding comments and blank lines)

4) LOC Table

The table below shows an example of how the LOC metric can be applied to different projects to estimate cost, effort, documentation requirements, and errors:

Project	LOC	Cost (in $k)	Efforts (Person-Months)	Documentation (Pages)	Errors
A	10,000	110	18	365	39
B	12,000	115	20	370	45
C	15,400	130	25	400	32

Q3. What are different metrics used for software measurement? Explain function Point-based estimation technique in detail

=>

Software Measurement Metrics

1) Software metrics are measurements used to assess different aspects of software projects, like quality, size, and productivity.
2) There are three main categories of software metrics:
    i) Product Metrics:
        a) These measure characteristics of the software product itself, such as size, complexity, quality, and reliability.
        b) They also help assess the maintainability of the product, indicating how easily the software can be updated or modified over time.
    ii) Process Metrics:
        a) These measure aspects of the software development process, like the rate of defects found during development, patterns in defect arrival, and the time taken for specific tasks.
        b) Additionally, process metrics help evaluate the efficiency of the development process by tracking resource usage and task completion rates.
    iii) Project Metrics:
        a) These measure project-related factors, such as the number of developers, project costs, schedules, and productivity levels.
        b) They also monitor risks and potential bottlenecks, allowing project managers to adjust timelines or resources as needed to ensure project success.

Function Point (FP) - Based Estimation Technique

1) The Function Point (FP) estimation technique is a **size-oriented** metric that measures the functionality of the software from the user's perspective, focusing on what the system does rather than how it's implemented.
2) It's used to estimate the "size" of the software, which helps in planning and resource allocation.
3) FP Table Example:

Measurement Parameter	Example
Number of External Inputs (EI)	Input screens and tables
Number of External Outputs (EO)	Output screens and reports
Number of External Inquiries (EQ)	Prompts and interrupts
Number of Internal Files (ILF)	Databases and directories
Number of External Interface Files (EIF)	Shared databases and routines

4) Function Point estimation helps project managers understand the size of a project based on its functionality, making it easier to allocate resources, estimate costs, and plan schedules.
5) It is especially useful when comparing different projects or when assessing productivity across various development environments.

Q4. Compute the function point value for a project with the following information domain characteristics:

1. **Number of user inputs: 32**
2. **Number of user output: 60,**
3. **Number of user enquiries: 24,**
4. **Number of files: 8,**
5. **Number of external interfaces: 0**

Assume that all complexity adjustment value are average and ∑ fi = 40

=> *This question was last asked in 2015*

Step 1: Calculate Unadjusted Function Points (UFP)

The formula for calculating UFP is:

$$\text{UFP} = (I \times W_I) + (O \times W_O) + (E \times W_E) + (F \times W_F) + (N \times W_N)$$

Where:

- I = Number of external inputs
- O = Number of external outputs
- E = Number of external inquiries
- F = Number of files
- N = Number of external interfaces

Given Data:

- Number of user inputs (I) = 32
- Number of user outputs (O) = 60
- Number of user inquiries (E) = 24
- Number of files (F) = 8
- Number of external interfaces (N) = 0

Complexity Weights:

- W_I (External Inputs) = 4
- W_O (External Outputs) = 5
- W_E (External Enquiries) = 4
- W_F (Internal Logical Files) = 10
- W_N (External Interface Files) = 7

Calculation:

Substituting the values into the UFP formula:

$$\text{UFP} = (32 \times 4) + (60 \times 5) + (24 \times 4) + (8 \times 10) + (0 \times 7)$$

Calculating each term:

- $32 \times 4 = 128$
- $60 \times 5 = 300$
- $24 \times 4 = 96$
- $8 \times 10 = 80$
- $0 \times 7 = 0$

Now summing these values:

$$\text{UFP} = 128 + 300 + 96 + 80 + 0 = 604$$

Step 2: Calculate Value Adjustment Factor (VAF)

The VAF is calculated using:

$$VAF = TDI \times 0.01 + 0.65$$

Where TDI is the Total Degree of Influence, which is given as:

$$TDI = \sum f_i = 40$$

Calculation:

Substituting into the VAF formula:

$$VAF = (40 \times 0.01) + 0.65 = 0.40 + 0.65 = 1.05$$

Step 3: Calculate Function Point Value

Finally, we calculate the Function Point value using:

$$FP = UFP \times VAF$$

Substituting our values:

$$FP = 604 \times 1.05 = 634.2$$

Thus, the computed function point value for the project is **634.2**.