Types of Maintenance

Corrective   Perfective   Adaptive   Preventive

Corrective Maintenance

or live program

- Maintenance occurs when testing ↑ reveals a fault or error in the program and this is corrected.

Q- What are the reasons for using corrective maintenance?

- To amend algorithms to eliminate errors

Adaptive Maintenance

- It is a term used for changes that are made to a program in response to specification changes

Q. What are the reasons for adaptive maintenance?

- In response to specification changes arising from changes to business rules or environment.

As a result of changes to requirements of available technology.

Perfective Maintenance

- Mainly deals with implementing new or changed user requirement.
- Involves making functional enhancements to the code
- This includes both the function and the efficiency of the code

and changing the functionalities as per the user's changing needs.

Preventive Maintenance:

- Involves performing of activities to avoid occurrence of errors.

Types of Error

Syntax Error Logical Error Run-Time Error

Syntax Error

- Are small gramitical mistakes, sometimes limited to a single character

- Program does not execute

- E.g: A missing semi-colon at the end of the line or an extra bracket at the end of a function may produce a syntax error.

Logical Error

- Errors in the logic of a program

- Program does not run as expected.

E.g: Adding two numbers

5 , 3 ADD= 5-3

Print ADD → 2

Run-time Error

- Program executes an invalid instruction. (out of bounds error)

- E.g: - Attempts to divide by zero
  - -ve number in square root

- Logical and Run-time errors may only show up under certain circumstances.

- Program execution comes to an unexpected halt or it goes into an infinite loop.

STATE TRANSITION DIAGRAM

MACHINE → Machines have states →

- ON , OFF
- Traffic lights
  - Red, green, yellow

Finite State Machine

- Any machine that has fixed number of states

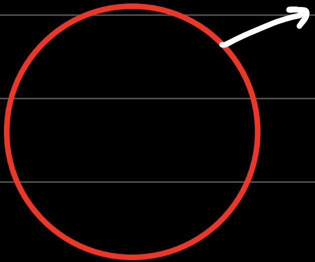
Q- What is transition?

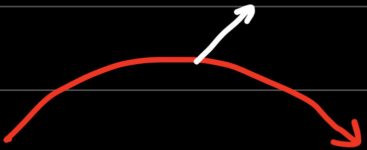
- Moving from one state to the other is known as transition.

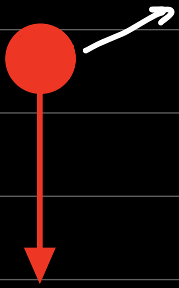
x ———————— x

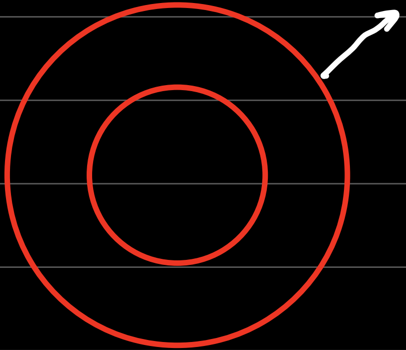
Purpose of State-transition diagrams

- Shows conditions need for an event or events to occur that cause a transition
- Outputs / actions carried out as a result of that transitio
- State transition diagrams show the behaviour of finite state machines.









Ways of Avoiding and Exposing Faults in a Program

Fault Avoidance

- Provision of comprehensive and rigorous program specification at the end of analysis phase

- Avoid making as many mistakes as possible and then find as many mistakes as possible before program goes live.

- At design stage, state-transition diagram, structure charts and pseudocode help in detecting faults

- At coding stage, the use of programming disciplines : info hiding, encapsulation and exception handling help to prevent faults

- Faults may be exposed when program goes live and at testing stage
  ↳ corrected during maintenance stage.