Section 12.3 - Program Testing and Maintenance

Layer 7: Application

Syllabus Content Section 12: Software Development

S12.3.1 Show understanding of ways of exposing and avoiding faults in programs

---

- the programmer has made a coding mistake
- the requirement specification was not drawn up correctly
- the soft ware designer has made a design error
- the user interface is poorly designed, and the user makes mistakes
- computer hardware experiences failure.

S12.3.2 Locate and identify the different types of errors

- syntax errors
- logic errors
- run-time errors

> Syntax errors:

- When source code does not obey rules of the language
- Compiler generates error messages
- Examples:
  - Misspell identifier when calling it
  - Missing punctuation – colon after if
  - Incorrectly using a built-in function
  - Argument being made does not match data type

> Logic errors:

- Program works but gives incorrect output
- Examples:
  - Out By One – when '>' is used instead of '>='
  - Misuse of logic operators

> Run-time errors:

- Source code compiles to machine code but fails upon execution (red lines show up in Python)
- When the program keeps running and you have to kill it manually

- Examples:
    - Division by 0
    - Infinite loop – will not produce error message, program will just not stop until forced to

S12.3.3 Correct identified errors

> Error

`Syntax error`: an error in which a program statement does not follow the rules of the language
`Logic error`: an error in the logic of the solution that causes it not to behave as intended
`Run-time error`: an error that causes program execution to crash or freeze

Corrective Maintenance is correcting identified errors
White-Box testing: making sample data and running it
through a trace table
Trace table: technique used to test algorithms; make sure
that no logical errors occur e.g.

> Adaptive Maintenance

- Making amendments to:
    - Parameters: due to changes in specification
    - Logic: to enhance functionality or more faster or both
    - Design: to make it more user friendly

S12.3.4 Show understanding of the methods of testing available and select appropriate data for a given method

- Including dry run, walkthrough, white-box, black-box, integration, alpha, beta, acceptance, stub

---

> Black box testing:

- Use test data for which reults already calculated & Compare result from program with expected results
- Testing only considers input and output and the code is viewed as being in a 'black box'

> White box testing

- Examine each line of code for correct logic and accuracy.
- May record value of variables after each line of code
- Every possible condition must be tested

> Stub testing

- Stubs are computer programs that act as temporary replacement for a called module and give the same output as the actual product or software.
- Important when code is not completed however must be tested so modules are replaced by stubs

> Dry run testing

- A process where code is manually traced, without any software used
- The value of a variable is manually followed to check whether it is used and updated as expected
- Used to identify logic errors, but not execution errors

> Walkthrough testing

- A test where the code is reviewed carefully by the developer's peers, managers, team members, etc.
- It is used to gather useful feedback to further develop the code.

> Integreation testing

- Taking modules that have been tested on individually and testing on them combined together
- This method allows all the code snippets to integrate with each other, making the program work.

> Alpha testing

- This is the testing done on software 'in-house', meaning it is done by the developers
- Basically another term for 'first round of testing'

> Beta testing

- This is the testing done on the software by beta users, who use the program and report any problems back to the developer.
- Basically another term for 'second round of testing'

> Acceptance testing:

- A test carried out by the intended users of the system: the people who requested the software.
- The purpose is to check that the software performs exactly as required.
- The acceptance criteria should completely be satisfied for the program to be released.

S12.3.5 Show understanding of the need for a test strategy and test plan and their likely contents

---

there are different ways to test data

1. Dry Run

2. Walkthrough
3. White-Box
4. Black-Box
5. Integration
6. Alpha
7. Beta
8. Acceptance
9. Stub

S12.3.6 Choose appropriate test data for a test plan

- Including normal, abnormal and extreme/boundary

Type	Description	Example
Normal / Typical	Data that should be allowed	12, 74, 92
Extreme Data / Boundary Data	Should be allowed but is at the start and end of the range allowed	5, 100
Erroneous / Invalid / Abnormal	Should not be allowed	4, 102, Six

S12.3.7 Show understanding of the need for continuing maintenance of a system and the differences between each type of maintenance

- Including perfective, adaptive, corrective

Maintenance = Making sure your program keeps working

> `Perfective Maintenance`: modifying a program to improve performance or maintainability

You made your program.
You gave it to the user

They like it but they want some small changes.
Like changing the layout of the log in screen
Or moving a button to a different location

It doesn't mean you start from the beginning and re-write your whole program, you just make the small change needed.

Usually these changes are cosmetic / about the way it looks
This is perfective maintenance

> `Adaptive Maintenance`: amending a program to enhance functionality or in response to specification changes

This is when you have to change your program to adapt to a new situation.

The new situation may be outside your control

Example. You make a Chess game, and in chess the King can only move one square. But now the World Chess Organisation said “we change the rules, a king can move 2 squares”

You have to perform adaptive maintenance to your program

Or if your calculator says the tax value is 17% but now the government say its 20%. You have to change your program to adapt.

> `Corrective Maintenance`: correcting identified errors

This is where your program is being used and even after you tested it, there are some errors.

You don’ want to / need to re-write the whole program again, you just let people download a patch or an update that fixes these errors.

This is corrective maintenance

S12.3.8 Analyse an existing program and make amendments to enhance functionality

---