UNIT 3

Date. _____
Page No. _____

Coding

* Coding is the process of transforming the design of a system into a computer language format.
* Coding is done by the coders or programmers who are independent people than the designers
* Software Implementation/Development/Coding is an imp step in SDLC
* It results in an executable version of the software
* Coding guidelines are defined by the organization
* different tools such as compilers, debuggers, and interpreters are used by the developers
* Programming languages such as C, C++, Python, etc are selected on the requirement of the problem.

Goals of Coding

1) To translate the design of system into a computer language
2) Cost of testing and maintenance can be significantly reduced with efficient coding
3) Making the program more readable and understandable

Date.
Page No.

Structured Programming

Structured Programming helps to develop programs that are easier to understand. In this the code will execute the instruction by instruction one after the other. It doesn't support the possibility of jumping from one instruction to some other with the help of any statement like GOTO etc. The instructions in this approach will be executed in a serial and structured manner.

In Structured Programming, a program has both static and dynamic structures.

The static structure is the systematic organization of the program statements. The dynamic structure refers to the order of instructions to be executed.

Dynamic structure may vary based on the data input.

Three important principles in structured programming are
Sequence
Repitition
Selection

Page No.

Structured program consist of well separated and structured modules. But the entry and exit in a structured program is a single-time event. It means that the program uses single entry and single exit elements. Therefore a structured program is well maintained, neat and clean program.

This is the reason why the structured programming approach is well accepted in the programming world.

Advantages
Easier to read and understand
User friendly
Easier to maintain
Mainly problem based instead of machine based
Development is easier & less effort and time
Easier to debug
Machine independent

Disadvantages
Machine independent → takes time to convert into machine code
Machine code is not same as for assembly Coding language dependent.

Date.
Page No.

Information Hiding

A software always containing data structure to store information about the problem domain

In practice, only some amount of information is used frequently. Thus, some information should be hidden or not made available.

An access function must be provided for obtaining the data.

For eg
In a student's record, he/she should not be given access to modify the fees paid attribute by anybody except the in-charge accountant.

Date.
Page No.

Programming Style.

The programming style must be helpful in writing simple and clear code. with no or less errors.

The goal of good programming style is to provide understandable, straightforward elegant code.

Some important factors to achieve this objective are as follows

1) Control constructs
It is good to use single entry/exit control statements such as if, while, or do while etc.

2) gotos
Unconditional goto statement must be avoided.

3) Information hiding
There should be support for information hiding. Only the access functions for the data structures should be made visible while hiding the data structure behind these functions.

Date.
Page No.

4) Nesting
A complex nesting or loops must be avoided.

5) Module Size
Large module size will not be cohesive and difficult to debug.

6) Module Interface
The interface to the module should not be very complicated. It should be easily understandable and debug.

7) Side effects
While executing one module there may be modification of value in other parameters which is not desirable. These is called side effects. This should be avoided or well documented

8) Robustness
Exceptional conditions due to incorrect input, the incorrect value of some variable should not crash the program. Meaningful messages should be displayed in such situations
Empty if-else, loops, and return values must be checked.

Date.
Page No.

The data source should be trustworthy

Importance to the error handling using exceptions must be given

If there are switch/case statements, they should be accompanied by default cases.

Documentation
Code documentation is used to give information about the software product to the people who are using it.

Comments play important role in software documentation.

The purpose of the module, description of different data structure, and test cases to be used is given using comment.

Manuals, readme files, and installation procedures can be given in the documentation.

Contact information about the developer and maintenance team is also mentioned in the documentation.

Date.
Page No.

Testing

Error: When there is a mistake in a code then it is called an error. Errors may be the syntax, semantics, hardware, testing errors etc.

Fault: The state of incorrect working of a system is known as a fault. There may be a fault in the interface or front end, security, the performance of the software etc.

Failure: The system leads to failure if there are multiple defects. If the defect is causing incorrect functioning of the system then the system is in failed condition.

Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.

It plays an imp role in developing quality software.

Testing is performed using relevant data and the software is executed to check the expected output.

Date.
Page No.

The software under test is known as SUT.

The sample data used for checking the behaviours of the SUT is known as a test case and test suites.

Test case: A test case is a set of test inputs and execution conditions of the software. It also mentions the expected outcome from the software under test.

Test suite: It is a group of related test cases. Test suites result in some specific behavior of the software.

Test cases and suites are manual or automatic.

Functional Testing

Functional Testing refers to testing whether the system meets the predetermined requirements.

The processing of information in different functions is verified using test cases is functional testing.

Date.
Page No.

Along with the functional requirements the functional testing methods address quality attributes such as reliability, maintainability, security, etc.

Structural Testing

Structural testing refers to the testing of the code and design by the software developers. Specifications of the code, input/output constraints.

Eg data structures, front end, menus, low-level components, etc.

The logical and syntax errors are identified in structural testing.

The most used structural testing for software is control flow testing.

Control flow testing tests the order of the program execution when conditions and loops statements are used.