**1. Computer Basics**

Computer basics encompass fundamental concepts about computers and their operations.

**2. How Memory Works**

Memory functions as a critical component in computing, enabling data storage and retrieval.

**3. RAM (Random Access Memory)**

RAM is a volatile memory used by computers for storing data that is currently being processed.

**4. Stack**

The stack is a region of memory used for managing function calls and local variables.

**5. Heap**

The heap is a memory space used for dynamic memory allocation, including objects and data structures.

**6. Pointers**

Pointers are variables that store memory addresses, facilitating indirect data manipulation.

**7. Variables**

Variables are named storage locations used to hold data values in computer programs.

**8. Objects**

Objects are instances of classes, representing real-world entities in object-oriented programming.

**9. Programming Basics**

Programming basics cover fundamental concepts for writing code and building software.

**10. Algorithms**

Algorithms are step-by-step procedures used to solve specific problems or perform tasks.

**11. Concept**

Concepts in programming refer to high-level ideas and abstractions used to model real-world scenarios.

**12. Components**

Components are modular parts of a program that contribute to its overall functionality.

**13. Values**

Values are data elements stored in variables, representing different types of information.

**14. Variables (in Programming)**

Variables allow programmers to store and manipulate data during program execution.

**15. Instruction**

An instruction is a basic command that directs the computer to perform a specific operation.

**16. Procedure**

A procedure is a sequence of instructions executed to accomplish a particular task.

**17. Selection / Conditional Statements (if/else)**

Conditional statements allow the program to make decisions based on certain conditions.

**18. Repetition / Loops (for, while, do-while)**

Loops enable the repetition of a block of code until a specified condition is met.

**19. Documentation**

Documentation involves providing comments and explanations to make code understandable and maintainable.

**20. Dart Programming Basics**

Dart is a modern programming language developed by Google for building web and mobile applications.

**21. Values & Data Types**

Dart supports various data types like integers, strings, booleans, and floating-point numbers.

22. Variables

Variables in Dart are used to store values and are statically or dynamically typed.

**23. Conditional Statements**

Dart offers if and switch statements for implementing conditional logic.

**24. Loops**

Dart provides for, while, and do-while loops for iterative operations.

**25. Comments**

Comments in Dart help developers document code for better understanding.

**26. Functions**

Functions in Dart encapsulate code for reusability and modularization.

**27. Imports**

Imports in Dart allow including code from external libraries or files.

**28. Enums**

Enums define a set of named constant values in Dart.

**29. Classes**

Classes are the foundation of object-oriented programming in Dart, enabling the creation of custom types.

**30. Interfaces (Abstract Classes)**

Interfaces, achieved through abstract classes, define contracts that classes must adhere to.

**31. Inheritance**

Inheritance enables the creation of new classes based on existing ones, promoting code reuse.

**32. Composition**

Composition involves constructing complex objects by combining simpler ones.

**33. Mixins**

Mixins allow the reuse of code across different class hierarchies in Dart.

**34. Generics**

Generics enable the creation of classes and functions that can work with different data types.

**35. Futures & Async/Await**

Futures facilitate asynchronous programming, and async/await simplifies handling asynchronous operations.

**36. Streams**

Streams provide a way to handle sequences of asynchronous data in Dart.

**37. Exceptions**

Exceptions handle runtime errors and abnormal conditions in Dart programs.

**38. Null**

Null represents the absence of a value in Dart.

**39. Null Safety**

Null safety is a feature in Dart that prevents null reference errors.