**Que.1**

Abstraction:

Abstraction is nothing but showing only essential features and hiding the critical inner features.

What to show and what not to show.

Two sides of same coin.

Examples:

ATM machine:

In ATM machine we retrieve the money from machine. We only put our credentials on screen but we don’t know how system internally works.

Driving the car:

We drive the without knowing the how engine works. Only essential features are shown to you like Accelerator, Break, Clutch, Steering wheel, Gears.

**Que.2**

Data abstraction in C++:

There are 7 access modifiers are there in C++:

Public

Private

Protected

Default

Final

Static

Abstract

For achieving the data abstraction in C++ we use Private and Protected.

For showing the data we use Public and Default access modifiers and for hiding the data we use Private and Protected access modifiers.

Data members (Data) should not be shown, Business functions (Methods) can be shown.

**Que.3**

Inheritance:

The process of creating a new class from existing class reusing its features is called inheritance.

Only public and protected data and methods are derived by child class from parent class.

It is one of the central pillars of the object oriented concepts.

It is also called as “is a” kind of relationship.

Inheritance follows “Generalization – Specialization” design principle.

There are total 5 types of Inheritance

1. Single level inheritance
2. Multi-level inheritance
3. Hierarchical inheritance
4. Multiple inheritance
5. Hybrid inheritance

Class supports only first three types of inheritance.

1. Single level inheritance: In this inheritance, a derived class is created from a single base class. E.g. Father and Son
2. Multi-level inheritance: In this inheritance, a derived class is created from another derived class. E.g. Grandfather, father and son
3. Hierarchical inheritance: In this inheritance, more than one derived classes are created from a single base class and further child classes act as parent classes for more than one child classes. E.g. Father with two or more sons.
4. Multiple inheritance: In this inheritance, a derived class is created from more than one base class. E.g. Son with two parents i.e. Mother and Father.
5. Hybrid inheritance: This is combination of more than one inheritance. E.g. when class A and B extends class C & another class D extends class A then this is a hybrid inheritance, because it is a combination of single and hierarchical inheritance.

Advantages of Inheritance

1. Reduce code redundancy.
2. Provides code reusability.
3. Reduces source code size and improves code readability.
4. The code is easy to manage and divided into parent and child classes.
5. Supports code extensibility by overriding the base class functionality within child classes.

Disadvantages of Inheritance

1. In Inheritance base class and child class, both are tightly coupled. Hence If you change the code of parent class, it will affect all the child classes.
2. In a class hierarchy, many data members remain unused and the memory allocated to them is not utilized. Hence it affects the performance of your program if you have not implemented inheritance correctly.

**Que.4**

UML:

It stands for Unified Modeling Language.

UML is not a Programming Language it is a general purpose modeling language.

The UML uses mostly graphical notations to express the OO analysis and design of software projects

The main aim of UML is to define standard overall view of system.

Diagrams in UML:

Structural Diagram.

Behaviour Diagram.

Structural Diagram: Class Diagram, Object Diagram.

Behaviour Diagram: Use Case, State Diagram, Activity Diagram.

+ For Public

* For Privare

# For Protected

/ For Derived

Que.5

Different types of UML diagram

* Use Case Diagram
* Class Diagram
* Sequence Diagram
* Collaboration Diagram
* State Diagram