**ASSIGNMENT 1**

MODULE 1

B1 WHAT IS SOFTWARE ENGNEERING?

Software engineering is the art of developing quality software on time and within budget.

B2 WHAT IS SDLLC?

A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

B3 WHAT IS SOFTWARE DEVELOPMENT METHODOLOGY?

A software development methodology/process, also referred to as the software development lifecycle is the set of principles or the rules that directs the development team to the successful completion of the project. The decision to choose a methodology depends upon the project requirements and time frame.

B4 WHAT IS AGILE METHOD?

Agile is a collection of principles used in software development and project management. Agile focuses on enabling teams to deliver work in small, workable increments, thus delivering value to their customers with ease. Evaluation of the requirements, plans, and results take place continuously. This helps the team in responding to changes in a quick manner.

The major principles of Agile are detailed in the Agile manifesto. Created in early 2001, the Agile manifesto details the different values and principles that embody the process. The manifesto states:

Individuals and Interactions OVER Process and Tools

Working Products OVER Comprehensive Documentation

Customer Collaboration OVER Contract Negotiation

Responding to Changes OVER Following a Plan

**B5** WHAT IS USE CASE?

A Use Case in Testing is a brief description of a particular use of the software application by an actor or user. Use cases are made on the basis of user actions and the response of the software application to those user actions. It is widely used in developing test cases at system or acceptance level.

**B6** WHAT IS ACTIVITY DIAGRAM?

ACTIVITY DIAGRAM is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The basic purpose of activity diagrams is to capture the dynamic behavior of the system. It is also called object-oriented flowchart.

**B7** WHAT IS SRS?

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

**B8** WHAT IS PROGRAMMING?

Put simply, programming isgiving a set of instructions to a computer to execute**.** If you’ve ever cooked using a recipe before, you can think of yourself as the computer and the recipe’s author as a programmer. The recipe author provides you with a set of instructions which you read and then follow.

**B9** WHAT IS OOPS?

An object-based programming language is one which easily supports object-orientation. Identifying objects and assigning responsibilities to these objects. Objects communicate to other objects by sending messages. Messages are received by the methods of an object

**B 10**  WRITE BASIC CONCEPTS OF OOPS

BASIC CONCEPTS OF OOPS ARE

OBJECT

CLASS

ENCAPSULATION

INHERITANCE

POLYMORPHISM

ABSTRACTION

**B 11** WHAT IS OBJECT?

An object is an entity or instance of a class. The objects are mostly the physical entity but it can be a logical entity as well. Each object has state and behaviours.

An object always model or depicts the real world entities like a mango, the Hero Passion, Toyota Fortuner, etc.

**B 12** WHAT IS A CLASS?

The class is a model or blueprint or prototype of an object that defines or specifies all the properties of the objects. Classes have the data and its associated function wrapped in it. The class defines the state and behaviours of an object.

**B 13** WHAT IS RDBMS?

**RDBMS** stands for *Relational Database Management System.* All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS. It is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd.

**B 14** WHAT IS SQL?

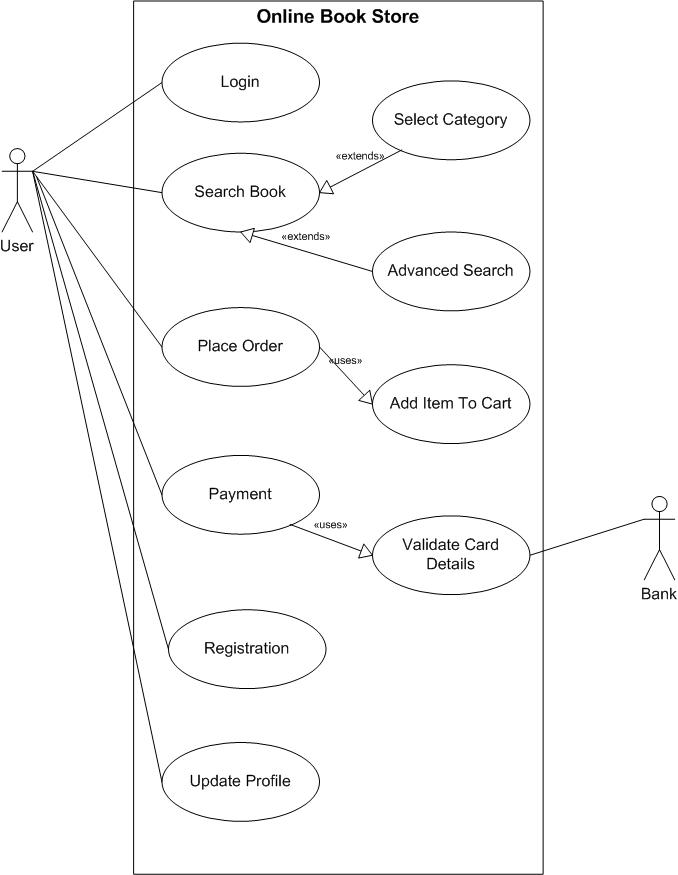
SQL stands for **S**tructured **Q**uery **L**anguage. SQL is a standard language for accessing databases. SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database. SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

**B 15** WRITE SQL COMMANDS

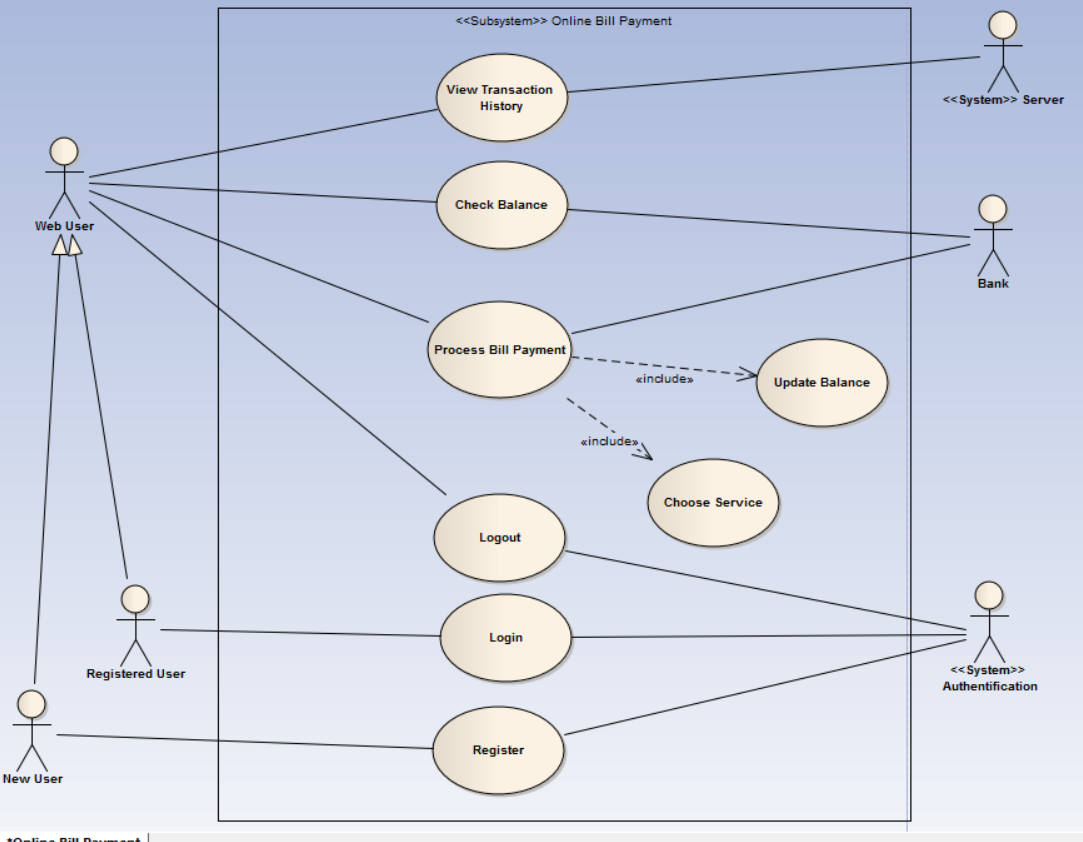
The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature −

* [**Data Definition Language(DDL)**](https://www.edureka.co/blog/sql-commands#DDL) – Consists of commands which are used to define the database. CREATE,DROP,ALTER AND TRUNCATE.
* [**Data Manipulation Language(DML)**](https://www.edureka.co/blog/sql-commands#DML) – Consists of commands which are used to manipulate the data present in the database.INSERT,UPDATE AND DELETE.
* [**Data Control Language(DCL)**](https://www.edureka.co/blog/sql-commands#DCL) – Consists of commands which deal with the user permissions and controls of the database system. GRANT AND REVOKE.
* [**Transaction Control Language(TCL)**](https://www.edureka.co/blog/sql-commands#TCL) – Consist of commands which deal with the transaction of the databases.COMMIT AND ROLLBACK.

**B 16** DRAW USE-CASE ON ONLINE BOOK SHOPPING.



**B 17** DRAW USE CASE on online bill payment system



**I1**  WRITE SDLC PHASES WITH BASIC INTRODUCTION

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

### Stage 1: Requirement gathering

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

### Stage 2: Requirement Analysis

### Once the requirement gathering and analysis is done the next step is to define and [document](https://www.softwaretestingmaterial.com/documentation-testing-in-software-testing/) the product requirements and get them approved by the customer. This is done through the SRS (Software Requirement Specification) document. SRS consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Design

It has two steps:  
HLD – High-Level Design – It gives the architecture of the software product to be developed and is done by architects and senior developers  
LLD – Low-Level Design – It is done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code

Stage 4 Coding

Developers of all levels (seniors, juniors, freshers) involved in this phase. This is the phase where we start building the software and start writing the code for the product. The outcome from this phase is Source Code Document (SCD) and the developed product.

Stage 5 Testing

When the software is ready, it is sent to the testing department where Test team tests it thoroughly for different defects. They either test the software manually or using automated testing tools depends on the process defined in [STLC (Software Testing Life Cycle)](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/) and ensure that each and every component of the software works fine.

Stage 6 Maintenance

After successful testing, the product is delivered/deployed to the customer for their use. Deployment is done by the Deployment/Implementation engineers. Once when the customers start using the developed system then the actual problems will come up and needs to be solved from time to time. Fixing the issues found by the customer comes in the maintenance phase.

**I2** EXPLAIN TYPES OF REQUIREMENT

A software requirement can be of 2 types:

* Functional requirements
* Non-functional requirements

**Functional Requirements:** These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

**Non-functional requirements:** These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.   
They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

**I3** STATE THE IMPORTANCE OF DESIGN PHASE.

software design is the process of setting a foundation for constructing your software structure code. This is what makes it the most important aspect of [software development](https://www.mindbowser.com/software-product-engineering-services/) yet there are many corner sights to it, which may give reasonable answers to the importance of software design. Whenever a software developer starts working on any type of new project, it is obvious that he would want to start scripting the code right away. Regardless of what his expertise is over the software are, if he builds software without laying out a plan for it, then he might be building a home without setting a strong foundation.

**I4** WHAT ARE THE TASKS PERFORMED IN CODING STAGE?

In this phase, developers start build the entire system by writing code using the chosen programming language. In the coding phase, tasks are divided into units or modules and assigned to the various developers. It is the longest phase of the Software Development Life Cycle process.

In this phase, Developer needs to follow certain predefined coding guidelines. They also need to use [programming tools](https://www.guru99.com/software-development-tools.html) like compiler, interpreters, debugger to generate and implement the code.

**I5** BRIEFLY EXPLAIN TESTING PHASE

Once the software is complete, and it is deployed in the testing environment. The testing team starts testing the functionality of the entire system. This is done to verify that the entire application works according to the customer requirement.

During this phase, QA and testing team may find some bugs/defects which they communicate to developers. The development team fixes the bug and send back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

**I6** EXPLAIN PHASES OF WATERFALL MODEL

The sequential phases in Waterfall model are −

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**I7** WRITE PHASES OF SPIRAL MODEL.

## Spiral Model Phases

It has four stages or phases: The planning of objectives, risk analysis, engineering or development, and finally review. A project passes through all these stages repeatedly and the phases are known as a Spiral in the model.

1. **Determine objectives and find alternate solutions –** This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.
2. **Risk Analysis and resolving –** In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.
3. **Develop and test:** This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.
4. **Review and planning of the next phase –** In this phase,the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

I8 WRITE AGILE MANIFESTO PRINCIPLES.

#### AGILE MANIFESTO PRINCIPLES ARE: 1. Individuals and interactions over processes and tools

This value of the Agile manifesto **focuses on giving importance to communication with the clients**. There are several things a client may want to ask and it is the responsibility of the team members to ensure that all questions and suggestions of the clients are promptly dealt with.

#### 2. Working product over comprehensive documentation

In the past, more focus used to be on proper documentation of every aspect of the project. There were several times when this was done at the expense of the final product. The Agile values dictate that **the first and foremost duty of the project team is completing the final deliverables** as identified by the customers.

#### 3. Customer collaboration over contract negotiation

Agile principles **require customers to be involved in all phases of the project**. The [Waterfall approach](https://kissflow.com/project/agile/traditional-vs-agile-project-management/) or Traditional methodologies only allow customers to negotiate before and after the project. This used to result in wastage of both time and resources. If the customers are kept in the loop during the development process, team members can ensure that the final product meets all the requirements of the client.

#### 4. Responding to change over following a plan

Contrary to the management methodologies of the past, Agile values are against using elaborate plans before the start of the project and continue sticking to them no matter what. Circumstances change and sometimes customers demand extra features in the final product that may change the [project scope](https://kissflow.com/project/project-scope-management/). In these cases, project managers and their **teams must adapt quickly in order to deliver a quality product and ensure 100% customer satisfaction**.

**I9** WHAT IS ACTOR IN USE CASE

Actor in ause case diagram is **any entity that performs a role** in one given system. This could be a person, organization or an external system and usually drawn like skeleton shown below.



**I10** HOW MANY KINDS OF NODES IN ACTIVITY DIAGRAM.WHICH?

An activity diagram consists of a number of nodes connected by arrowed lines (directed edges). There are three main types of node: action nodes, object nodes, and control nodes. Action nodes represent some work being carried out and are shown as a rectangle with rounded corners. The rectangle typically contains the name of the action: other information may optionally be added that we won't consider here. Object nodes, shown as rectangles with square corners, represent an object type, for example, a business entity such as an invoice that may be passed from one activity to another. Object nodes are generally only shown when some special consideration needs to be given to the objects being passed: routine use of business objects by activities is just assumed. The “flow” along the activity edges is either of control tokens, indicating the transfer of control from one activity to another, or of objects. Since the edges are directed, there is no assumed direction of flow, although diagrams are conventionally laid out so that the main flows are either left to right or top to bottom.

Control nodes are abstract activity nodes that coordinate flows in an activity diagram and come in various subtypes. An *initial* node is represented by a filled circle and represents a starting point for the activity diagram. There can be more than one initial node. There are two kinds of *final* node. An *activity final node* is shown as a target symbol, and an activity diagram can have more than one of these. As soon as an activity final node receives a token, the entire activity immediately terminates even if there are other control tokens active in other paths. A [*flow final node*](https://www.sciencedirect.com/topics/computer-science/flow-final-node)*,* shown as an “X” in a circle, destroys any token that reaches it, without affecting any other concurrent flows.

A *decision* node, denoted by a diamond shape, has one input edge and one or more output edges. A token arriving on the input edge will leave on only one of the output edges (i.e., exclusive-or). To control this, the output edges have guard conditions, denoted inside square brackets. The guard conditions should be complete and disjoint to ensure that token flow is not inadvertently halted or duplicated. A default branch can be provided by specifying it with the guard condition [else].

**I11** WHAT IS ENCAPSULATION?

**Encapsulation** is one of the fundamentals of OOP (object-oriented programming). It refers to the bundling of data with the methods that operate on that data. Encapsulation is used to hide the values or state of a structured data object inside a class, preventing unauthorized parties’ direct access to them. Publicly accessible methods are generally provided in the class (so-called getters and setters) to access the values, and other client classes call these methods to retrieve and modify the values within the object.[[1]](https://press.rebus.community/programmingfundamentals/chapter/encapsulation/#footnote-1899-1)

**I12** WHAT IS INHERITANCE?

In [object-oriented programming](https://www.computerhope.com/jargon/o/oop.htm), **inheritance** refers to the ability of an [object](https://www.computerhope.com/jargon/o/object.htm) to take on one or more characteristics from other [classes](https://www.computerhope.com/jargon/c/class.htm) of objects. The characteristics inherited are usually instance [variables](https://www.computerhope.com/jargon/v/variable.htm) or member [functions](https://www.computerhope.com/jargon/f/function.htm). An object that inherits these characteristics is known as a subclass. The object it inherits them from is known as a [superclass](https://www.computerhope.com/jargon/s/superclass.htm).

**I13** WHAT IS ABSTRACTION?

Abstraction is one of the [key concepts](https://stackify.com/oops-concepts-in-java/) of object-oriented programming (OOP) languages. Its main goal is to handle complexity by hiding unnecessary details from the user. That enables the user to implement more complex logic on top of the provided abstraction without understanding or even thinking about all the hidden complexity.

That’s a very generic concept that’s not limited to object-oriented programming. You can find it everywhere in the real world.

**I14** WHAT IS POLYMORPHSIM?

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A real-life example of polymorphism, a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person posses different behavior in different situations. This is called polymorphism. Polymorphism is considered as one of the important features of Object Oriented Programming.

**In C++ polymorphism is mainly divided into two types:**

* Compile time Polymorphism
* Runtime Polymorphism

**I15** WHAT IS SQL?

SQL is the most common language for extracting and organising data that is stored in a relational database. A database is a table that consists of rows and columns. SQL is the language of databases. It facilitates retrieving specific information from databases that are further used for analysis. Even when the analysis is being done on another platform like Python or R, SQL would be needed to extract the data that you need from a company’s database.

**I16** WRITE SQL COMMANDS IN DETAILS.

SQL commands are mainly categorized into four categories as:

1.DDL – Data Definition Language

2.DQl – Data Query Language

3.DML – Data Manipulation Language

4.DCL – Data Control Language

### **DDL (Data Definition Language):**

[DDL](https://www.geeksforgeeks.org/features-of-structured-query-language-sql/) or Data Definition Language actually consists of the SQL commands that can be used to define the database schema.

List of DDL commands:

* [**CREATE**](https://www.geeksforgeeks.org/sql-create/): This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
* [**DROP**](https://www.geeksforgeeks.org/sql-drop-truncate/): This command is used to delete objects from the database.
* [**ALTER**](https://www.geeksforgeeks.org/sql-alter-add-drop-modify/)**:** This is used to alter the structure of the database.
* [**TRUNCATE**](https://www.geeksforgeeks.org/sql-drop-truncate/)**:** This is used to remove all records from a table, including all spaces allocated for the records are removed.
* [**COMMENT**](https://www.geeksforgeeks.org/sql-comments/): This is used to add comments to the data dictionary.
* [**RENAME**](https://www.geeksforgeeks.org/sql-alter-rename/)**:** This is used to rename an object existing in the database.

### **DQL (Data Query Language):**

**DQL** statements are used for performing queries on the data within schema objects.

List of DQL:

* [**SELECT**](https://www.geeksforgeeks.org/sql-select-clause/)**:** It is used to retrieve data from the database.

### **DML(Data Manipulation Language):**

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements

List of DML commands:

* [**INSERT**](https://www.geeksforgeeks.org/sql-insert-statement/) : It is used to insert data into a table.
* [**UPDATE**](https://www.geeksforgeeks.org/sql-update-statement/)**:** It is used to update existing data within a table.
* [**DELETE**](https://www.geeksforgeeks.org/sql-delete-statement/) : It is used to delete records from a database table.
* [**LOCK:**](https://www.geeksforgeeks.org/sql-lock-table/) Table control concurrency.
* **CALL:** Call a PL/SQL or JAVA subprogram.
* **EXPLAIN PLAN:** It describes the access path to data

### **DCL (Data Control Language):**

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

* [**GRANT:**](https://www.geeksforgeeks.org/mysql-grant-revoke-privileges/) This commandgives users access privileges to the database.
* [**REVOKE:**](https://www.geeksforgeeks.org/difference-between-grant-and-revoke/)This command withdraws the user’s access privileges given by using the GRANT command.

**I17** WHAT IS A JOIN?

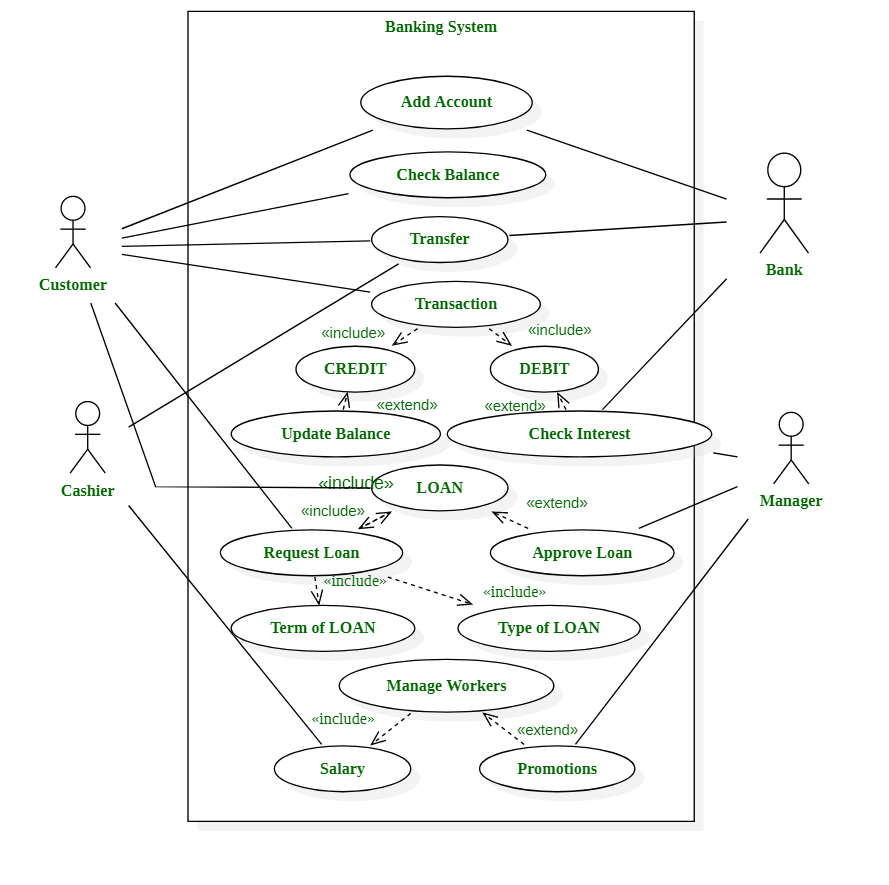
The SQL **Joins** clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

**I18** WRITE TYPES OF JOIN

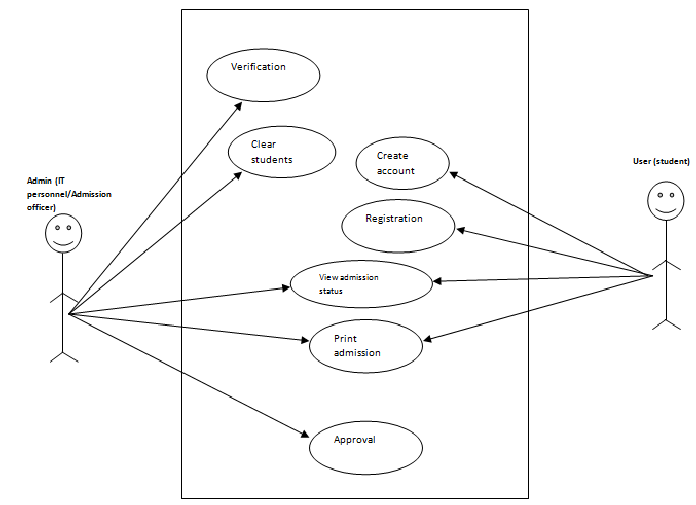
There are different types of joins available in SQL −

* [INNER JOIN](https://www.tutorialspoint.com/sql/sql-inner-joins.htm) − returns rows when there is a match in both tables.
* [LEFT JOIN](https://www.tutorialspoint.com/sql/sql-left-joins.htm) − returns all rows from the left table, even if there are no matches in the right table.
* [RIGHT JOIN](https://www.tutorialspoint.com/sql/sql-right-joins.htm) − returns all rows from the right table, even if there are no matches in the left table.
* [FULL JOIN](https://www.tutorialspoint.com/sql/sql-full-joins.htm) − returns rows when there is a match in one of the tables.

**I19** DRAW USECASE ON ONLINE PAYMENT TRANSFER VIA BANK TO BANK.



**I20** DRAW USE CASE ON STUDENT REGISTRATION PROCESS ON UNIVERSITY.



**A1** EXPLAIN PHASES OF SDLC IN DETAIL.

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

### Stage 1: Requirement gathering

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

### Stage 2: Requirement Analysis

### Once the requirement gathering and analysis is done the next step is to define and [document](https://www.softwaretestingmaterial.com/documentation-testing-in-software-testing/) the product requirements and get them approved by the customer. This is done through the SRS (Software Requirement Specification) document. SRS consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Design

It has two steps:  
HLD – High-Level Design – It gives the architecture of the software product to be developed and is done by architects and senior developers  
LLD – Low-Level Design – It is done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code

Stage 4 Coding

Developers of all levels (seniors, juniors, freshers) involved in this phase. This is the phase where we start building the software and start writing the code for the product. The outcome from this phase is Source Code Document (SCD) and the developed product.

Stage 5 Testing

When the software is ready, it is sent to the testing department where Test team tests it thoroughly for different defects. They either test the software manually or using automated testing tools depends on the process defined in [STLC (Software Testing Life Cycle)](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/) and ensure that each and every component of the software works fine.

Stage 6 Maintenance

After successful testing, the product is delivered/deployed to the customer for their use. Deployment is done by the Deployment/Implementation engineers. Once when the customers start using the developed system then the actual problems will come up and needs to be solved from time to time. Fixing the issues found by the customer comes in the maintenance phase.

**A2** ACCORDING TO YOU WHICH IS THE MOST CREATIVE AND CHALLENGING PHASE OF SDLC?

The most creative and challenging phase of the system life cycle is system design, which refers to the technical specifications that will be applied in implementing the candidate system. It also includes the construction of programmers and program testing.

It has the following stages −

* Acquiring hardware and software, if necessary
* Database design
* Developing system processes
* Coding and testing each module

The final report prior to implementation phase includes procedural flowcharts, record layout, report layout and plan for implementing the candidate system. Information on personnel, money, hardware, facility and their estimated cost must also be available. At this point projected cost must be close to actual cost of implementation.

**A3** WHO ARE THE PEOPLE INVOLVED IN THE PHASES OF WATERFALL MODEL.

Waterfall is an old-school method, with an old-school command structure.

Larger teams take control of the project, in some cases can consist of more than 15 people in non-interchangeable groups. These individuals involved abide by a strict hierarchy, with the lead role going to the project manager.

**A4** EXPLAIN THE TEAM REQUIREMENT GATHERING CONCERNING SDLC.

The most important phase of the SDLC is the requirement gathering and analysis phase because this is when the project team begins to understand what the customer wants from the project. During the requirements gathering sessions, **the project team meets with the customer to outline each requirement in detail** To make sure that all the steps mentioned above are appropriately executed, clear, concise, and correct requirements must be gathered from the customer. The customer should be able to define their requirements properly and the business analyst should be able to collect it the same way the customer is intending it to convey.

Many a time it is not possible that requirement gathering is done efficiently by business analysts from the customer. This might be due to dependency on many people related to the expected end product, tools, environment, etc. Thus, it is always a good idea to involve all the stakeholders who could influence or could be influenced by the end product.

**A5** WHAT ARE THE PROBLEMS FACED IN WATERFALL MODEL?

The major disadvantages of the Waterfall Model are as follows −

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.
* Adjusting scope during the life cycle can end a project.
* Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

**A6** WRITE APPLICATION OF ITERATIVE AND INCREMENTAL MODEL.

**Application of iteration Model:**

There is various software development models in [SDLC](http://er.yuvayana.org/definition-of-software-development-life-cyclephases-of-sdlc/) approach available and used by all software application development industries. Each software development application has different requirements based on internal and external factors. There have some situations where the software development has highly required the use of iteration model is more suitable and handle the problem very smartly:

* **When** the requirements of the complete system are clearly defined and understood by the development expertise.
* Major requirements of the system must be defined initially; however, some functionalities or requested enhancements may evolve with time to time.
* Resources with required skill set are not available and are planned to be used on contract basis for specific iterations.
* A new technology is being used and is being learnt by the development team while working on the project.
* There are some high risk features and goals which may change in the future.
* There is a time to the market constraint.

**A7** WRITE PROS AND CONS OF ITERATIVE AND INCREMENTAL MODEL.

|  |  |
| --- | --- |
| **Pros of iteration model** | **Cons of iteration model** |
| Progress of developed project can be measured. | System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle. |
| Some working functionality can be developed quickly and early in the life cycle. | More resources may be required. |
| Testing and debugging during smaller iteration is easy. | Not suitable for smaller projects. |
| Less costly to change the scope/requirements. | Defining increments may require definition of the complete system. |
| Issues, challenges & risks identified from each increment can be utilized/ applied to the next increment. | Project.s progress is highly dependent upon the risk analysis phase. |
| Results are obtained early and periodically. | Although cost of change is lesser but it is not very suitable for changing requirements. |
| Parallel development can be planned. | More management attention is required. |
| Risks are identified and resolved during iteration; and each iteration is an easily managed milestone. | Management complexity is more. |
| Easier to manage risk – High risk part is done first. | End of project may not be known risk. |
| With every increment operational product is delivered. | Highly skilled resources are required for risk analysis. |
| During life cycle software is produced early which facilitates customer evaluation and feedback. |  |
| It supports changing requirements. |
| Initial Operating time is less. |
| Better suited for large and mission-critical projects. |

**A8** WRITE APPLICATIONS OF SPIRAL MODEL

**Application of SDLC spiral model:**

Various software application development models available in market those are used by the software industries. Each software development application has different requirements based on internal and external factors. Similar for the spiral model, this modeling approach is very widely used in the software industry as it is in synch with the natural development process of any product i.e. learning with maturity and also involves minimum risk for the customer as well as the development firms. Following are the typical uses of Spiral model:

* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* When costs there are a budget constraint and risk evaluation is important.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* For medium to high-risk projects development.
* Customer is not sure of their requirements which are usually the case.
* Significant changes are expected in the product during the development cycle.

**A9** EXPLAIN WORKING METHODOLOGY OF AGILE MODEL AND ALSO WRITE PROS AND CONS.

AGILE METHODOLOGY is a process of software development (such as other software development methodologies – waterfall model, V-model, iterative model, etc.), however, the Agile development model is also a type of incremental model. Software develops in incremental, rapid cycles. In English, Agile means ‘the ability to move quickly and easily’ and respond to change rapidly – this is an important aspect of Agile software development.

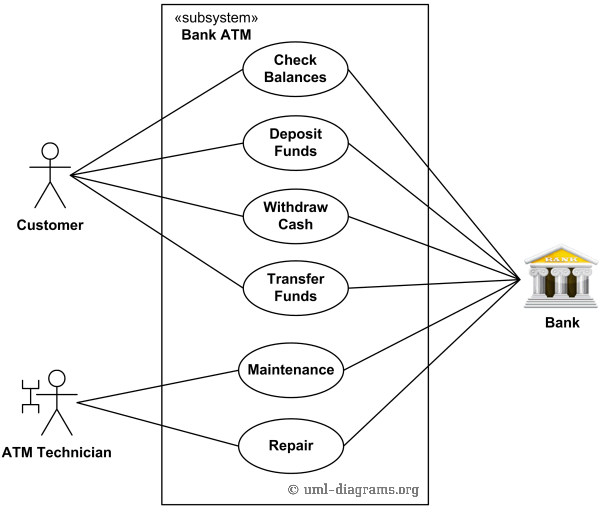
**Advantages of Agile Methodology :**

1. In Agile methodology the delivery of software is unremitting.
2. The customers are satisfied because after every Sprint working feature of the software is delivered to them.
3. Customers can have a look of the working feature which fulfilled their expectations.
4. If the customers has any feedback or any change in the feature then it can be accommodated in the current release of the product.
5. In Agile methodology the daily interactions are required between the business people and the developers.
6. In this methodology attention is paid to the good design of the product.
7. Changes in the requirements are accepted even in the later stages of the development.
8. An Agile/Scrum approach can improve organizational synergy by breaking down organizational barriers and developing a spirit of trust and partnership around organizational goals.

**Disadvantages of the Agile Methodology :**

1. In Agile methodology the documentation is less.
2. Sometimes in Agile methodology the requirement is not very clear hence it’s difficult to predict the expected result.
3. In few of the projects at the starting of the software development life cycle it’s difficult to estimate the actual effort required.
4. Because of the ever-evolving features, there is always a risk of the ever-lasting project.
5. For complex projects, the resource requirement and effort are difficult to estimate.

**A 10** CREATE USE CASE FOR ATM MACHINE.



**A11** create use case for e-shopping user module.



**A12** WHAT IS THE DIFFERENCE BETWEEN SOFTWARE PRODUCT AND SOFTWARE PROJECT.

# Difference between Project and Product

1. Project :  
Also known as a software project comprises the steps involved in making a product before it is actually available to the market. The project can be handled by people which are as less as one person to the involvement of a lot of people (over 100). These are usually assigned by an enterprise and are undertaken to form a new product that has not already been made.

2. Product :  
The study of products is a part of Software engineering. The software is built by developers on requests from the customer. After the customer is satisfied with the development process, he launches the software by manufacturing it. This can be a problem-solving software or computer based system. This is the result of a project. The software project, when completed, is called a product after it is available to the market for usage.

**A13** EXPLAIN POLYMORPHISM WITH EXAMPLE.

In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A real-life example of polymorphism, a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So the same person posses different behavior in different situations. This is called polymorphism. Polymorphism is considered as one of the important features of Object Oriented Programming.

In C++ polymorphism is mainly divided into two types**:**

* Compile time Polymorphism
* Runtime Polymorphism

**A14** EXPLAIN ABSTRACTION WITH EXAMPLE.

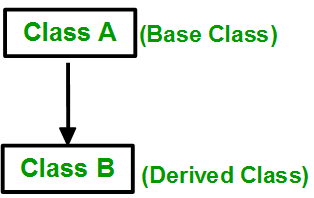
Data abstraction is one of the most essential and important feature of object oriented programming in C++. Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

Consider a real life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car. This is what abstraction is.

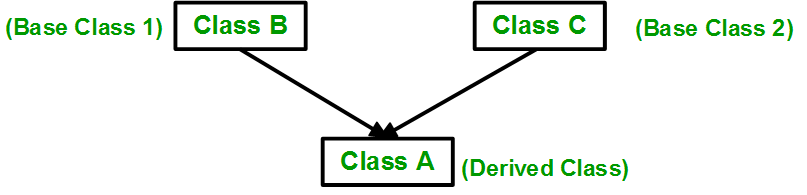
**A15** EXPLAIN TYPES OF INHERITANCE WITH EXAMPLE.

The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important feature of Object Oriented Programming.   
Sub Class: The class that inherits properties from another class is called Sub class or Derived Class.   
Super Class: The class whose properties are inherited by sub class is called Base Class or Super class.   
Types of Inheritance in C++

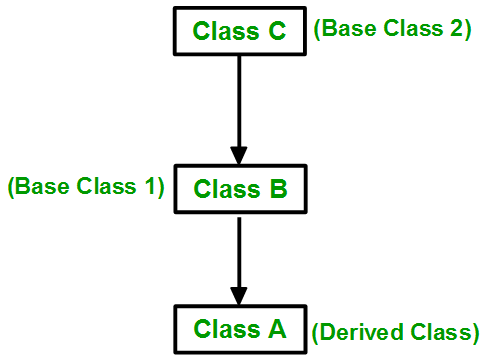
**1. Single Inheritance**: In single inheritance, a class is allowed to inherit from only one class. i.e. one sub class is inherited by one base class only.



**2. Multiple Inheritance:** Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes. i.e one **sub class** is inherited from more than one **base classes**.



3. **Multilevel Inheritance**: In this type of inheritance, a derived class is created from another derived class.



**A16** WHAT IS INNER JOIN?

The INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. The query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, column values for each matched pair of rows of A and B are combined into a result row.

**A17** EXPLAIN LEFT AND RIGHT JOIN WITH EXAMPLE.

LEFT JOIN: This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of join. The rows for which there is no matching row on right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN.Syntax:

SELECT table1.column1,table1.column2,table2.column1,....  
FROM table1   
LEFT JOIN table2  
ON table1.matching\_column = table2.matching\_column;  
  
  
table1: First table.  
table2: Second table  
matching\_column: Column common to both the tables.

**Example Queries(LEFT JOIN)**:

SELECT Student.NAME,StudentCourse.COURSE\_ID   
FROM Student  
LEFT JOIN StudentCourse   
ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**Output**:



RIGHT JOIN: RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of join. The rows for which there is no matching row on left side, the result-set will contain *null*. RIGHT JOIN is also known as RIGHT OUTER JOIN.Syntax:

SELECT table1.column1,table1.column2,table2.column1,....  
FROM table1   
RIGHT JOIN table2  
ON table1.matching\_column = table2.matching\_column;  
  
  
table1: First table.  
table2: Second table  
matching\_column: Column common to both the tables.

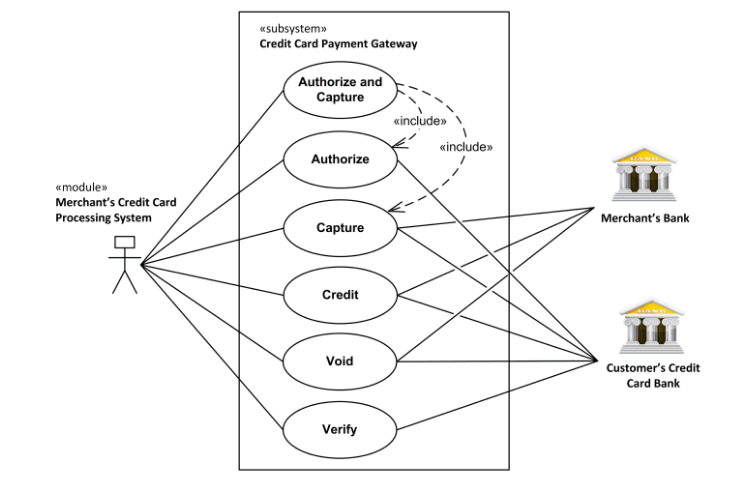
**Example Queries(RIGHT JOIN)**:

SELECT Student.NAME,StudentCourse.COURSE\_ID   
FROM Student  
RIGHT JOIN StudentCourse   
ON StudentCourse.ROLL\_NO = Student.ROLL\_NO;

**Output:**



**A18** DRAW USECASE ON ONLINE SHOPPING PRODUCT USING PAYMENT GATEWAY.



**A19** DRAW USE CASE ON PROPERTY PORTALWEB BASED PROJECT.

## 