1. **What is SDLC**

SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment and ongoing maintenance and support.

There are a number of different development models.

**2.What is software testing?**

Software testing is a process used to identify the correctness, completeness and quality of developed computer software.

**3.What is agile methodology?**

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirement. In agile the tasks are divided to time boxes to deliver specific features for a release.

**4.What is SRS?**

A Software Requirement Specification is a complete description of the behaviour of the system to be developed.it includes a set of use cases that describe all of the interactions that the users will have the software. Use cases are also known as functional requirement.

In addition to use cases, the SRS also contains non-functional requirement. This standard describes possible structures, desirable contents and qualities of a software requirements Specification.

**5.What is OOPS?**

An object based programming language is one which easily supports object orientation

Programming is like writing. If you can write a demonstration you can make a program. But actually programmer’s lots of wisdom, lots of knowledge about programming and lots of experience.

**6.Write Basic Concepts of OOPS**

Identifying objects and assigning responsibilities to these objects. Objects communicate by the methods of an objects. An object is like a black box. The internal details are hidden. Object of a program interact by sending messages to each other.

**7.What is object**

An object is anything to which a concept applies. This is the unit of object oriented programming. That is both data are bundled as a unit called object.

**8.What is class**

It is Blue print for an object

A class represents an abstraction of the object and abstracts the

properties and behaviour of object.

An object is a particular instance of a class which has actual existence and

there can be many objects (or instances) for a class.

**9.What Is encapsulation**

Encapsulation is the practice of including in an object everything

it needs hidden from other objects.

Encapsulation is placing the data and the functions that work on that

data in the same place.

Encapsulation in Java is the process of wrapping up of data

(properties) and behaviour of an object into a single unit;

and the unit here is a Class.

We can expose our operations hiding the details of what is needed to

perform that operation.

**10.What is inheritance.**

To Access property of one class to another class.

Java supports single-parent, multiple-children inheritance

and multilevel inheritance (Grandparent-> Parent -> Child) for classes

and interfaces. Java supports multiple inheritances (multiple parents,

single child) only through interfaces.

**11.What Is polymorphism.**

Poly refers to many. That is a single function or an operator functioning in

many ways different upon the usage is called polymorphism.

Same method name but having different functionality types of polymorphism.

*The ability to change form is known as polymorphism.*

# **Two types of polymorphism**.

⚫ Compile time polymorphism(Overloading)

⚫ Runtime polymorphism(Overriding)

**Overloading**

The same method name (method overloading) or operator

symbol (operator overloading) can be used in different

contents.

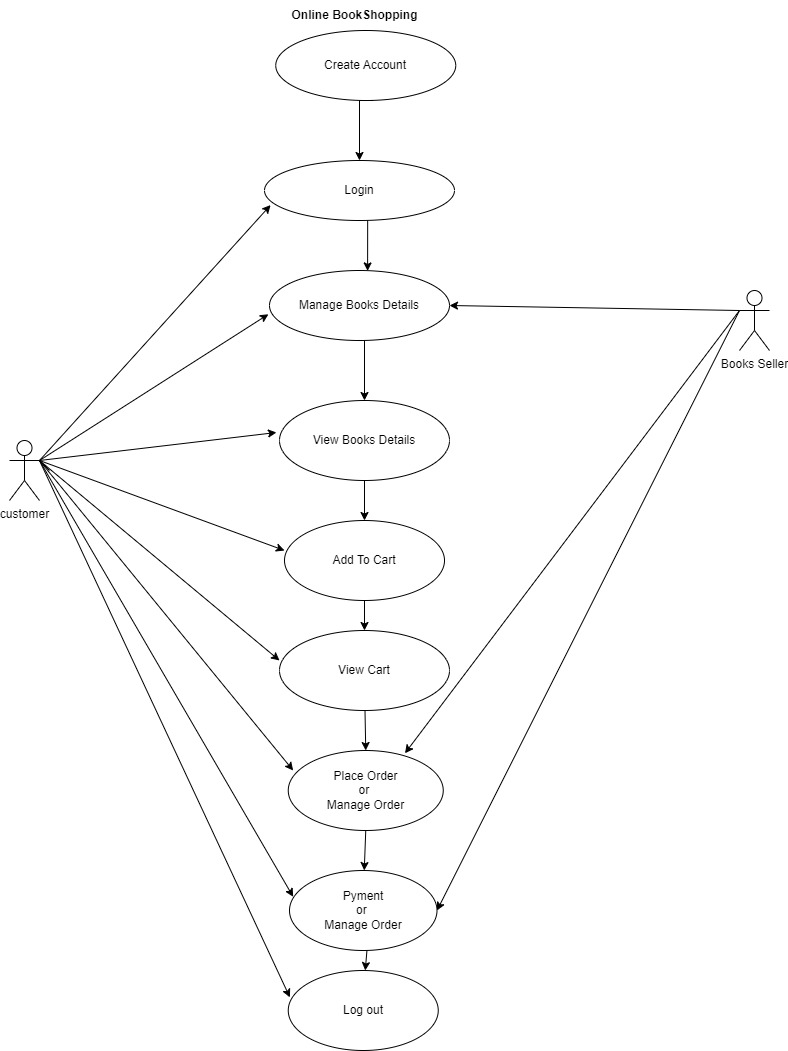
In method overloading, multiple methods having same name can

appear in a class, but with different signature.

**Overriding**

Same method name same argument into different class and inheritance compulsory.

**12.Draw Use case on Online Book Shopping**



**13.Draw use case on online bill payment system**

**14.Write SDLC phases with basic introduction**

1. **Requirements Collection/Gathering**
2. **Analysis**
3. **Design**
4. **Implementation**
5. **Testing**
6. **Maintenance**

**1.Requirements Collection**

⚫ Types of Requirements:

⚫ Functional Requirements: describe system services or

functions.

⚫ Compute sales tax on a purchase

⚫ Update the database on the server

⚫ Non-Functional Requirements: are constraints on the system or

the development process.

⚫ Non-functional requirements may be more critical than

functional requirements.

⚫ If these are not met, the system is useless.

**2.Analysis**

**⚫** This phase starts with the requirement document delivered by the

requirement phase and maps the requirements into architecture.

⚫ The architecture defines the components, their interfaces and

Behaviours.

**3.Design**

⚫ Design Architecture Document.

⚫ The Design team can now expand upon the information established in

the requirement document.

⚫ The requirement document must guide this decision process.

**4.Implementation**

⚫ In the implementation phase, the team builds the components either

from scratch or by composition.

⚫ For example, a component may be narrowly designed for this particular

system, or the component may be made more general to satisfy a

reusability guideline.

**5.Testing**

**⚫** Simply stated, quality is very important. Many companies have not

learned that quality is important and deliver more claimed

functionality but at a lower quality level.

⚫ Quality is a distinguishing attribute of a system indicating the

degree of excellence.

⚫ Regression Testing

⚫ Internal Testing

⚫ Unit Testing

⚫ Application Testing

⚫ Stress Testing

6.Maintenance

⚫ Software maintenance is phases in the System

Development Life Cycle (SDLC), as it applies to software development.

The maintenance phase is the phase which comes after deployment of

the software into the field.

⚫ updating all analysis, design and user documentation.

**15. Explain Phase of the Waterfall model**

⚫ Requirements are very well documented, clear and fixed.

⚫ Product definition is stable.

⚫ The project is short.

**Pros**

⚫ Simple and easy to understand and use

⚫ Phases are processed and completed one at a time

⚫ Easy to arrange tasks.

⚫ Process and results are well documented.

**Cons**

⚫ High amounts of risk and uncertainty.

⚫ Cannot accommodate changing requirements

⚫ Not a good model for complex and object-oriented projects.

⚫ Poor model for long and ongoing projects

⚫ No working software is produced until late in the life cycle.

⚫ Adjusting scope during the life cycle can end a project.

**16. Write phases of spiral model**

⚫ Spiral Model is very widely used in the software industry as it is in synch

with the natural development process of any product.

⚫ For medium to high-risk projects.

⚫ Long-term project commitment because of potential changes to

economic priorities as the requirements change with time.

⚫ Significant changes are expected in the product during the

development cycle.

**⚫ Pros**

⚫ Changing requirements can be accommodated.

⚫ Allows for extensive use of prototypes

⚫ Requirements can be captured more accurately.

⚫ Users see the system early.

**⚫ Cons**

⚫ End of project may not be known early.

⚫ Not suitable for small or low risk projects and could be expensive for

small projects.

⚫ Process is complex

⚫ Spiral may go indefinitely**.**

**17.Write agile manifesto principles.**

**1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**

**2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.**

**3.** **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**

**4. Business people and developers must work together daily throughout the project.**

**5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**

**6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**

**7. Working software is the primary measure of progress.**

**8.Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**

**9.Continuous attention to technical excellence and good design enhances agility.**

**10.Simplicity–the art of maximizing the amount of work not done–is essential.**

**11.The best architectures, requirements, and designs emerge from self-organizing teams.**

**12.At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.**

**18.Expain working methodology of agile model and also write pros and cons.**

⚫ Agile SDLC model is a combination of iterative and incremental

process models with focus on process adaptability and customer

satisfaction by rapid delivery of working software product.

⚫ Agile Methods break the product into small incremental builds.

⚫ Agile thought process had started early in the software

development and started becoming popular with time due to its

flexibility and adaptability.

⚫ **Pros**

⚫ Is a very realistic approach to software development

⚫ Promotes teamwork and cross training.

**⚫** Functionality can be developed rapidly and demonstrated

⚫ Suitable for fixed or changing requirements

Little or no planning required

⚫ Easy to manage

⚫ Gives flexibility to developers

⚫ **Cons**

⚫ More risk of sustainability, maintainability and extensibility.

⚫ An overall plan, an agile leader and agile PM practice is a must without

which it will not work.

⚫ Strict delivery management dictates the scope, functionality to be

delivered, and adjustments to meet the deadlines.

⚫ Depends heavily on customer interaction, so if customer is not

clear, team can be driven in the wrong direction.

⚫ There is very high individual dependency, since there is minimum

Documentation generated.

**19.Draw use case on online shopping product using COD**

****

**20. Draw use case on online shopping product using payment gateway**

****