**1.WHAT IS SDLC?**

**ANS:-The** software development life cycle (**SDLC) is the** process **of planning**, **writing,** modifying, **and** maintaining **software**. **Developers use** the methodology as they design and write modern software for computers, cloud **deployment,** mobile phones, **video games**, **and** more. Adhering **to** the **SDLC** methodology **helps to optimize** the final

**outcome**.

In IT, the term **"life cycle" was first used in the 1950s** and 1960s to describe the stages involved in developing a **new** computer **system**, but **it is now** commonly **used** to refer **to** all **stages** in the production of **any type of software**.

**2.WHAT IS SOFTWARE TESTING?**

**ANS:-Software** Testing **is a** method **to check whether the** actual **software product** matches expected **requirements** and to **ensure** that **software product is Defect free**. It **involves execution of**

software**/system components** using **manual** or automated **tools** to evaluate **one or** more **properties of interest**. The **purpose of software** testing **is to identify errors,** gaps **or** missing **requirements** in **contrast** to **actual** requirements.

**3.WHAT IS AGILE METHODOLOGY?**

**ANS:**-Agile **methodology is** a **project management framework** that **breaks** projects **down into several** dynamic **phases**, commonly known **as sprints**.

The **Agile framework is** an iterative **methodology**. **After every** sprint,

teams **reflect** and look **back** to **see** if there **was** anything that could be

**improved so they** can **adjust their** strategy **for** the **next sprint**.

**4.WHAT IS SRS**

**ANS:-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**.

**5.WHAT IS OOPS**

ANS**:-As** the **name** suggests, **Object-Oriented Programming or** OOPs

**refers** to languages **that use objects** in programming. **Object-oriented**

**programming aims** to **implement real-world entities like inheritance**,

hiding, polymorphism, etc in programming. The **main aim** of OOP is **to**

**bind together** the data and the functions that **operate** on them **so** that

no **other part of the** code **can access** this **data except that** function.

**6.WRITE BASIC CONCEPTS OF OOPS.**

**ANS:-**

• **OBJECT**

• **CLASS**

• ENCAPSULATION

• INHERITANCE

• **POLYMORPHISM**

..OVERRIDING

• .OVERLOADING

• ABSTRACTION

7.WHAT **IS OBJECT:-It is** a basic unit of Object-Oriented

Programming **and represents the real-life entities**. An **Object is an**

**instance of a Class**. When a **class is** defined, **no** memory is allocated

but when **it is instantiated (i.e.** an object **is created) memory is**

**allocated**. An **object has an identity**, **state**, and **behavior**. Each object

contains **data and code** to **manipulate** the **data**. **Objects** can interact

**without having** to **know details of each other's** data or **code,** it **is**

sufficient **to know the** type of message accepted and type **of response**

**returned** by the **objects**.

**For** example "Dog" is a real**-life Object**, **which has some**

characteristics **like** color**, Breed**, **Bark**, **Sleep**, **and Eats**.

**8.WHAT IS CLASS**

**ANS:-A** class **is a user-defined** data **type**. **It consists** of data **members**

**and member functions**, **which** can **be accessed and used** by **creating**

**an** instance **of that class. It represents the set of properties or**

**methods** that **are** common to all **objects of** one **type**. A class **is** like a

**blueprint for** an **object**.

**9. WHAT IS ENCAPSULATION**

**ANS:**-Encapsulation is defined **as** the wrapping up **of data** under a

**single** unit. It **is the mechanism** that binds together code and **the** data

it manipulates. In Encapsulation, the variables **or** data of **a** class are

**hidden from** any **other class** and **can be accessed** only **through any**

**member** function **of** their class in **which they are declared**. As **in**

**encapsulation**, **the data** in **a class is hidden from** other **classes, so it is**

**also known as** data-hiding.

**10.WHAT IS INHERITANCE**

**ANS:-Inheritance is** an **important pillar of** OOP(**Object**-Oriented

Programming**).** The capability **of** a class to **derive properties and**

**characteristics from another class is called Inheritance**. When **we write**

a **class**, **we inherit properties from other classes**. **So when we create** a

**class**, **we do** not **need to write** all the **properties and** functions again

**and again**, **as these** can **be inherited** from **another class that**

**possesses** it. **Inheritance allows** the **user to reuse** the code **whenever**

**possible** and **reduce its** redundancy.

**11.WHAT IS POLYMORPHISM**

**ANS:-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. **For example**, **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.

**12.DRAW USECASE ON ONLINE BOOK SHOPPING**

**ANS:-**

ONLINE **BOOK** SHOPPING

**ADMIN**

**LOGIN**

**ADD CATEGORY**

ADD ITEM

**MANAGE** ITEM

**MANAGE ORDER**

RAGISTRATION

VIEW **ITEM**

**MAKE** ORDER

**MAKE PAYMENT**

**13.DRAW USECASE ON ONLINE BILL PAYMENT**

**SYSTEM(PAYTM)**

**USER**

**ANS**:**-**

**USER**

**ONLINE** BILL **PAYMENT**

**LOGIN**

UPDATE BANK BALANCE

**BILL** PAYING **MENU**

FILL DETAIL

**MAKE PAYMENT**

PAYMENT **SUCCESSFULLY**

**LOGOUT**

**14.WRITE SDLC PHASES WITH BASIC INTRODUCTION**

**ANS:-**

**14.1.REQUIREMENTS COLLECTION/GETHERING**:**-ESTABLISH**

**CUSTOMER** NEEDS

**14.2.ANALYSIS:-MODEL** AND SPECIFY **REQUIREMENTS**

**14.3.DESIGN:-MODEL AND** SPECIFY A SOLUTION

**14.4.IMPLEMENTATION:-CONSTRUCT** A SOLUTION IN

SOFTWARE

**14.5.TESTING:**-VALIDATE THE SOLUTION AGAINST THE

**REQUIREMENTS**

**14.6.MAINTENANCE:**-REPAIR DEFECTS **AND** ADAPT THE

SOLUTION **TO THE NEW REQUIREMENTS**

**15.EXPLAIN PHASES OF THE WATERFALL MODEL**

**ANS**:**-**

**1.Requirements** analysis **and specification phase: The aim** of **this**

**phase is to understand the exact requirements of the customer** and **to**

document **them** properly. Both **the customer and** the **software**

developer **work** together **so as** to document **all** the **functions**,

performance, and interfacing requirement **of** the **software. It** describes

**the "what"** of **the system** to be produced **and not** "**how**.**"**In **this phase, a**

**large** document called Software Requirement Specification (**SRS**)

**document is** created which contained **a** detailed **description of what**

the **system will** do in the common language.

**2.Design Phase: This phase aims** to **transform the requirements**

**gathered in the SRS** into **a suitable** form **which permits further** coding

in **a programming** language. **It defines the overall software**

architecture together with high level and detailed design. All **this work**

**is** documented **as a Software** Design Document (SDD).

**3.Implementation and unit testing: During this** phase, design **is**

**implemented**. If the SDD **is complete**, **the implementation or** coding

**phase proceeds smoothly**, **because all** the **information needed by**

**software developers is contained** in the **SDD**.

**4. Integration and System Testing:** This **phase is** highly **crucial as**

**the** quality of **the end product is determined by the effectiveness of the**

**testing carried** out. The **better output will** lead to **satisfied** customers,

**lower maintenance costs**, **and** accurate **results**. Unit **testing**

**determines the efficiency of individual modules. However, in** this

**phase, the modules are tested for their interactions with** each other

**and with** the **system**.

**5.Operation and maintenance phase:**Maintenance is the **task**

performed by **every user once** the **software has been delivered to** the

**customer**, installed**, and operational**.

**16.WRITE PHASE OF SPIRAL MODEL:-**

**ANS:-**

• **PLANNING:**DETERMINATION OF

OBJECTIVES,**ALTERNATIVES AND CONSTRAINTS**

• **RISK ANALYSIS:ANALYSIS** OF ALTERNATIVES AND

IDENTIFICATION**/RESOLUTION** OF **RISKS**

• **ENGINEERING**:**DEVELOPMENT** OF THE "NEXT

LEVEL"PRODUCT

• **CUSTOMER EVALUATION:ASSESSMENT OF** THE **RESULT**

OF ENGINEERING

**17.WRITE AGILE MANIFESTO PRINCIPLES**

**ANS**:**-**

• Our highest **priority is** to **satisfy the customer through early** and

**continuous delivery of** valuable **software**.

• **Welcome** changing **requirements**, **even late** in **development**.

Agile **processes harness** change **for** the **customer's competitive**

advantage.

• **Deliver** working **software frequently**, **from** a couple **of weeks** to **a**

**couple of** months, with a preference **to the** shorter **timescale**.

• **Business** people and **developers must work together daily**

**throughout the project**.

**Build** projects **around motivated individuals**. **Give them the**

**environment** and **support** they **need**, **and trust them** to **get the** job

done.

**18.EXPLAIN WORKING METHODOLOGY OF AGILE MODEL AND**

**ALSO WRITE PROS AND CONS**.

**ANS**:**-**

**18.1METHODOLOGY:-**

• AGILE **METHODS BREAK** THE PRODUCT INTO **SMALL**

INCREMENTAL **BUILDS**

• THESE **BUILDS** ARE PROVIDED IN INTERATIONS.

• EACH INTERACTION TYPICALLY LASTS FROM **ABOUT** ONE

TO THREE **WEEKS**.

**18.2.PROS:-**

• **CHANGING REQUIREMENTS CAN BE ACCOMMODATED**

• **ALLOWS** FOR EXTENSIVE **USE** OF **PROTOTYPES**

• **REQUIREMENTS CAN BE** CAPTURED **MORE** ACCURATELY

• USERS SEE THE SYSTEM EARLY

**18.3.CONS:-**

• MANAGEMENT IS **MORE** COMPLEX

• END OF PROJECT **MAY NOT BE KNOWN EARLY**.

• **PROCESS IS** COMPLEX

• **SPIRAL** MAY **GO INDEFINITELY**

**19.DRAW USECASE ON ONLINE SHOPPING PRODUCT USING**

**COD**.

**ANS:-**

USER

ONLINE SHOPPING PRODUCT

LOGIN

VIEW **MAIN MENU**

**SELECT** PRODUCT

**GO** TO **CART**

**ADJUST SIZE** AND **QUANTITY**

SELECT **CASH** ON DELIVERY

CONFIRM ORDER

**TRACK ORDER**

**20.DRAW USE CASE ON ONLINE SHOPPING PRODUCT USING**

**PAYMENT GATEWAY.**

**ANS:-**

**USER**

ONLINE SHOPPING PRODUCT

**LOGIN**

VIEW MAIN MENU

SELECT PRODUCT

GO TO CART

ADJUST SIZE AND QUANTITY

SELECT PAYMENT GATEWAY

CONFIRM ORDER

TRACK ORDER