Name :Manoj Kumar gunda_1

SECTION 1: THEORY QUESTIONS

Q1. Explain the Software Development Life Cycle (SDLC) and its importance in software engineering.

Answer:
An Operating System (OS) is system software that manages computer hardware and provides essential services to application programs. It acts as an intermediary between users and hardware, enabling efficient resource utilization. Without an operating system, users would not be able to interact with hardware devices directly.

One of the core responsibilities of an OS is process management. It handles the execution of multiple processes by scheduling CPU time and managing process states. Memory management is another key responsibility, ensuring that programs are allocated sufficient memory while preventing conflicts between processes.

The operating system also manages file systems, allowing users to create, read, write, and delete files. It maintains directory structures and ensures data is stored persistently on storage devices. Device management is handled through drivers that allow the OS to communicate with hardware such as printers and disks.

Security and user management are also vital OS functions. Operating systems control access permissions and isolate user activities to prevent unauthorized actions. Overall, the OS ensures stability, efficiency, and security in computing systems.

Q2. Describe the role of Database Management Systems (DBMS) in modern applications.

Answer:
A Database Management System, or DBMS, is software that allows users to create, manage, and manipulate databases efficiently. It acts as an interface between the user and the database, ensuring that data is stored, retrieved, and updated securely. Modern applications heavily rely on DBMS to handle large volumes of structured data.

One of the key roles of a DBMS is data organization. It stores data in tables with rows and columns, making it easier to manage relationships between different data entities. DBMS also provides data integrity by enforcing constraints such as primary keys and foreign keys, which prevent invalid data entry.

Another important function of DBMS is concurrency control. Multiple users can access the database simultaneously without causing data inconsistency. Transaction management

ensures that operations follow ACID properties, maintaining reliability even during system failures.

Security is another critical role of DBMS. It provides authentication, authorization, and access control mechanisms to protect sensitive data. Backup and recovery features ensure data can be restored in case of hardware or software failures.

Overall, DBMS simplifies data handling, improves performance, ensures data accuracy, and supports scalable application development.

Q3. Explain the concept of Operating Systems and their core responsibilities.

Answer:
An Operating System (OS) is system software that manages computer hardware and provides essential services to application programs. It acts as an intermediary between users and hardware, enabling efficient resource utilization. Without an operating system, users would not be able to interact with hardware devices directly.

One of the core responsibilities of an OS is process management. It handles the execution of multiple processes by scheduling CPU time and managing process states. Memory management is another key responsibility, ensuring that programs are allocated sufficient memory while preventing conflicts between processes.

The operating system also manages file systems, allowing users to create, read, write, and delete files. It maintains directory structures and ensures data is stored persistently on storage devices. Device management is handled through drivers that allow the OS to communicate with hardware such as printers and disks.

Security and user management are also vital OS functions. Operating systems control access permissions and isolate user activities to prevent unauthorized actions. Overall, the OS ensures stability, efficiency, and security in computing systems.

Q4. Discuss the fundamentals of Computer Networks and their applications.

Answer:
The Software Development Life Cycle, commonly known as SDLC, is a structured process used by software engineers to design, develop, test, and maintain software systems. It provides a systematic approach that ensures software is developed efficiently and meets user requirements. SDLC typically begins with the requirement analysis phase, where stakeholders gather functional and non-functional requirements to understand the problem domain clearly.

The next phase is system design, where architects define system architecture, data flow, and technology stack. This stage plays a crucial role in defining how the software will function internally. After design, the implementation phase starts, where developers write code based on the approved design documents using selected programming languages and frameworks.

Testing follows implementation and ensures that the software works as expected. Various testing techniques such as unit testing, integration testing, and system testing are performed to identify defects. Once the software passes testing, it is deployed to the production environment where users can access it.

The final phase is maintenance, which involves fixing bugs, improving performance, and adding new features over time. SDLC is important because it improves project planning, reduces risks, enhances software quality, and ensures timely delivery. By following SDLC, organizations can build reliable and scalable software systems.

Q5. Explain Cloud Computing and its service models.

Answer:
Cloud Computing is a technology that allows users to access computing resources such as servers, storage, and applications over the internet. Instead of owning physical infrastructure, users can rent resources on demand from cloud service providers. This model reduces costs and increases flexibility.

Cloud computing offers scalability, allowing organizations to increase or decrease resources based on demand. It also improves availability, as cloud providers offer high uptime through distributed data centers. Data backup and disaster recovery are easier in cloud environments.

There are three main service models in cloud computing. Infrastructure as a Service (IaaS) provides virtualized hardware resources. Platform as a Service (PaaS) offers development platforms without managing infrastructure. Software as a Service (SaaS) delivers ready-to-use applications through web browsers.

Cloud computing is widely used in web hosting, data analytics, application development, and enterprise systems, making it a foundational technology in modern IT.

SECTION 2: CODING / LOGIC QUESTIONS

Q6. Implement a basic JWT authentication simulation in Python.

Answer:

import base64

```
import json
import time



def verify_token(token):
    try:
        decoded_bytes = base64.b64decode(token.encode())
        data = json.loads(decoded_bytes.decode())
        current_time = int(time.time())
        if current_time > data["expires"]:
            return False, "Token expired"
        return True, data["user"]
    except Exception as e:
        return False, str(e)

def main():
    user = "student1"
    token = generate_token(user)
    print("Generated Token:", token)

    is_valid, result = verify_token(token)
    if is_valid:
        print("Authenticated user:", result)
    else:
        print("Authentication failed:", result)

if __name__ == "__main__":
    main()
```

Q7. Write a Python program to perform CRUD operations using a dictionary.

Answer:

```
database = {}


def create_record(user_id, name, age):
    if user_id in database:
        print("Record already exists")
    else:
        database[user_id] = {"name": name, "age": age}


def read_record(user_id):
    return database.get(user_id, "Record not found")


def update_record(user_id, name=None, age=None):
    if user_id in database:
        if name:
            database[user_id]["name"] = name
        if age:
            database[user_id]["age"] = age
    else:
        print("Record not found")


def delete_record(user_id):
    if user_id in database:
        del database[user_id]
    else:
        print("Record not found")
```

```
def display_all():
    for key, value in database.items():
        print(key, value)

create_record(1, "Alice", 22)
create_record(2, "Bob", 25)
update_record(1, age=23)
delete_record(2)
display_all()
```

Q8. Develop a Python program to read and write data to a file with error handling.

Answer:

```
def write_file(filename, content):
    try:
        with open(filename, "w") as file:
            file.write(content)
    except IOError:
        print("Error writing to file")
```

Q9. Implement stack operations using Python lists.

Answer:

```
class Stack:
    def __init__(self):
        self.items = []

    def push(self, item):
```

```
        self.items.append(item)

    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        return "Stack is empty"

    def peek(self):
        if not self.is_empty():
            return self.items[-1]
        return None

    def is_empty(self):
        return len(self.items) == 0

    def size(self):
        return len(self.items)

stack = Stack()
stack.push(10)
stack.push(20)
print(stack.pop())
print(stack.peek())
print(stack.size())
```

Q10. Write a Python program to validate user input for login credentials.

Answer:

```
def validate_login(username, password):
```

```
    if not username or not password:
        return False
    if len(password) < 6:
        return False
    return True

def main():
    users = {"admin": "admin123"}
    username = input("Enter username: ")
    password = input("Enter password: ")

    if username in users and users[username] == password:
        if validate_login(username, password):
            print("Login successful")
        else:
            print("Invalid input format")
    else:
        print("Authentication failed")

if __name__ == "__main__":
    main()
```

SECTION 3: PROGRAMMING LANGUAGE QUESTIONS

Q11. Explain variables and data types in Python.

Answer:
Variables in Python are used to store data values that can be referenced later in a program. Python is dynamically typed, meaning the data type is determined at runtime. Common data types include integers, floats, strings, booleans, lists, tuples, dictionaries, and sets. This flexibility allows developers to write concise and readable code.

Q12. Describe Object-Oriented Programming concepts in Python.

Answer:
Object-Oriented Programming in Python is based on concepts such as classes, objects, inheritance, encapsulation, and polymorphism. Classes act as blueprints for objects, while objects represent real-world entities. OOP improves code reusability, modularity, and maintainability.

Q13. Explain the use of loops in Python.

Answer:
Loops in Python are used to execute a block of code repeatedly. The for loop is commonly used for iterating over sequences, while the while loop runs as long as a condition is true. Loops reduce code duplication and improve efficiency.

Q14. What are functions in Python and why are they important?

Answer:
Functions are reusable blocks of code that perform specific tasks. They improve code organization, reduce redundancy, and make programs easier to test and maintain. Functions can accept parameters and return values.

Q15. Explain Python data structures with examples.

Answer:
Structure the data in the good maner