SkillCortex AI

Redefining Education Through Innovation

Java™

FUNDAMENTAL QUEST

Q 1. What is the difference between JDK and JRE?

Ans. The JDK (Java Development Kit) is a complete pa
developers use to build Java applications. It inclu
tools, libraries, and a compiler required for writing
Java code.
On the other hand, the JRE (Java Runtime Environ
designed for running Java applications. It provid
environment along with necessary class libraries
contain development tools like a compiler.

Q 2. What are the advantages of using Java?

Ans. Here are some key advantages of Java:

- **Platform Independence:** Java programs can
system equipped with a Java Virtual Machine
ensuring cross-platform compatibility.
- **Enhanced Security:** Java incorporates a stro
framework that safeguards users against pot
code.
- **Object-Oriented Approach:** Java follows an
programming paradigm, making code more
reusable, and easier to maintain.
- **High Reliability:** Java is designed to be a stal

FUNDAMENTAL QUES

Q 3. What are the different components of the Java

Ans. The Java platform is a comprehensive software that enables the development and execution of applications. It comprises the following main co

- **Java Virtual Machine (JVM):** The core comp executes Java bytecode, making Java progr platform-independent.
- **Java Runtime Environment (JRE):** Provides t necessary libraries and runtime support for r applications.
- **Java Development Kit (JDK):** A complete too includes the JRE, compiler, and essential deve tools required for creating Java programs.

Q 4. What are the different types of control stateme

Ans. Java control statements are classified into three

- **Decision-making statements:** if, if-else, and
- **Looping statements:** while, do-while, and for
- **Jump statements:** break, continue, and retur

FUNDAMENTAL QUES

Q 5. What are the different types of data types in Ja

Ans. Java data types are categorized into two main primitive data types and non-primitive data typ

1. Primitive Data Types

These are the basic built-in data types in Java t simple values. They include:

- boolean – Stores true or false values.
- byte – A small integer type (8-bit).
- short – A 16-bit integer type.
- int – A 32-bit integer type (commonly used numbers).
- long – A 64-bit integer type for larger values
- float – A 32-bit floating-point number for de
- double – A 64-bit floating-point number, us decimal calculations.
- char – A 16-bit Unicode character type.

2. Non-Primitive Data Types

These are more complex data types that are de primitive types. They include:

- String – Represents a sequence of characte
- Array – A collection of elements of the same

FUNDAMENTAL QUEST

Q 6. What are the different types of Java exceptio

Ans. Java exceptions are categorized into two types

- **Checked Exceptions:** These must be declar method signature or handled using a try-c not managed properly, the compiler genera

```
try {
    File file = new File("myfile.txt");
    Scanner scanner = new Scanner(file)
    // Do something with the scanner
} catch (IOException e) {
    System.out.println("File not found"
}
```

- **Unchecked Exceptions**: These do not requi in the method signature. They can be throv method, and the compiler does not enforc them.

```
try {
    File file = new File("myfile.txt")
    Scanner scanner = new Scanner(file
    // Do something with the scanner
} catch (IOException e) {
    throw new RuntimeException(e);
}
```

FUNDAMENTAL QUEST

Q 7. What are the different types of Java classes a interfaces?

Ans. Java classes are mainly of two types:

- **Normal Classes:** Regular classes that can h methods, and constructors.
- **Abstract Classes:** Classes that cannot be in and are used as a base for other classes.

Java interfaces are categorized into:

- **Normal Interfaces:** Contain abstract metho class must implement.
- **Marker Interfaces:** Do not contain methods used to indicate a specific property or beha class.

Q 8. What are the different types of Java libraries a frameworks?

Ans. A Java library is a collection of reusable classes Examples include:

- Apache Commons
- Google Guava
- Joda-Time
- Joda-Time
- JUnit

FUNDAMENTAL QUEST

Q 9. What are the different types of Java threads?

Ans. Java threads are categorized into two types:

- **User Threads:** Created by users or applicatio high-priority threads. The JVM waits for all us complete before terminating.
- **Daemon Threads:** These provide services to u run in the background. They are low-priority t JVM terminates them once all user threads fi

Q 10. What are the different types of Java networking

Ans. Java networking is mainly classified into two typ

- **Client-Server Networking:** A model where a application requests services from a server c which processes and responds accordingly.
- **Peer-to-Peer (P2P) Networking:** A model wh applications communicate directly with each a central server.

OBJECT ORIENTED PROGRAMMING

Q 1. What is the difference between Procedural Prog OOP?

Ans.

- **Procedural Programming:** Uses a top-down where programs are structured as a series of operate on data. It focuses on procedures an
- **Object-Oriented Programming (OOP):** Uses approach, where programs are designed usi represent real-world entities. It focuses on da encapsulated within objects.

Q 2. What are the core concepts of OOP?

Ans. The four main concepts of OOP are:

1. **Abstraction:** Hides unnecessary details and s the important features. Example: A car driver to use the steering and pedals without knowi engine works.
2. **Encapsulation:** Keeps data and methods tog class and restricts direct access to the data. bank account class allows balance changes through deposit and withdraw methods.
3. **Inheritance:** Allows one class to inherit prope

OBJECT ORIENTED PROGRAMMING

Q 3. What is the difference between Overloading and

Ans.

- **Overloading:** Allows multiple methods to sha name within a class, but they must have diffe lists.
- **Overriding:** Occurs when a subclass provides implementation for a method that already ex superclass, using the same method signature

Q 4. What is the difference between Static and Dyna Binding?

Ans. Static binding and dynamic binding are two way resolving method calls in object-oriented progra (OOP).

- **Static Binding:** The method to be called is de compile time. It is commonly used for static c virtual methods.
- **Dynamic Binding:** The method to be called is at runtime, allowing polymorphism through v methods.

Feature	Static Binding	Dynamic Bi

OBJECT ORIENTED PROGRAMMING

Q 5. What is the difference between an Abstract Cla Interface?

Ans. The following table highlights the key differences abstract classes and interfaces:

Feature	Abstract Class	
Can be instantiated?	No	
Can have abstract methods?	Yes	
Can have non-abstract methods?	Yes	
Can have state (fields/variables)?	Yes	
Can be extended by other classes?	Yes (extends)	
Can be implemented by other classes?	Yes (extends)	Yes

Q 6. Why doesn't Java support Multiple Inheritance?

Ans. Java does not support multiple inheritance beca lead to a number of problems, including:

OBJECT ORIENTED PROGRAMMING

Q 1. When should you use an Interface and an Abstr Java?

Ans. Both abstract classes and interfaces are used fo Java, but they serve different purposes:

- **Abstract Class:** Used when you want to share behavior among multiple related classes. It c abstract (without a body) and non-abstract well as instance variables (state). It cannot b
- **Interface:** Used when you want to define a co multiple unrelated classes must follow. It can (instance variables) and must be implement concrete class.

Q 2. What are the challenges of using OOP in Java?

Ans. Some challenges of using Object-Oriented Progr in Java include:

- Complexity: Large systems with multiple objec relationships can be difficult to manage.
- Overhead: Objects take up memory and requ management, leading to additional overhead
- Testing: Objects need to be tested both indivi

DATA STRUCTURES
ALGORITHMS

Q 1. What is the difference between an Array and a L

Ans. Arrays and linked lists are two common data stru
with its own advantages:

- Arrays are best when fast access to elements
maintaining order are important.
- Linked Lists are ideal when frequent insertions
are needed.

Feature	Array	Linked Lis
Data Storage	Contiguous memory	Non-contiguous
Access Efficiency	High (direct indexing)	Low (sequential
Insertion/Deletion	Slow (shifting required)	Fast (just pointer
Order of Data	Important	Not import

Q 2. Explain the concept of a Hash Table.

Ans. A hash table is a data structure that stores key-v

DATA STRUCTURES ALGORITHMS

Q 3. What is the difference between an Array and a L

Ans. The time complexity of BST operations depends o
height. In a balanced BST, operations are efficien
while in a skewed BST, they can degrade to O(n).

Operation	Time Complexity
Search	O(log n)
Insert	O(log n)
Delete	O(log n)
Inorder Traversal	O(n)
Preorder Traversal	O(n)
Postorder Traversal	O(n)

DATA STRUCTURES ALGORITHMS

Q 4. What is the difference between Breadth-First Se Depth-First Search (DFS)?

Ans. BFS and DFS are two common graph traversal al different exploration strategies.

Feature	BFS (Breadth-First Search)	DFS (Dept
Exploration	Visits all nodes at the current level before moving deeper	Goes as d before b
Time Complexity	O(V + E)	O
Space Complexity	O(V) (stores all nodes at a level)	O(V) (store in wo
Use Cases	Shortest path finding, discovering all reachable nodes	Finding al puzzles

OBJECT ORIENTED PROGRAMMING

Q 5. What is a Priority Queue, and where is it used?

Ans. A priority queue is a data structure where each e
a priority, and elements are processed based on
rather than the order they were added. The highe
lowest) priority element is always retrieved first.

Example Application:

A priority queue is used in task scheduling for op
systems. Tasks are assigned priorities, ensuring t
priority tasks (like system-critical processes) are
before lower-priority ones.

Q 6. What is Dynamic Programming, and where is it

Ans. Dynamic Programming (DP) is a problem-solving
that solves complex problems by breaking them
overlapping subproblems and solving each subp
once using memoization or tabulation.

Example:

The Knapsack Problem is a classic DP problem. G
with weights and values, and a knapsack with lim
capacity, the goal is to determine the maximum
can fit within the weight limit. DP helps efficiently

OBJECT ORIENTED PROGRAMMING

Q 7. What is a Priority Queue, and where is it used?

Ans. A HashSet in Java is backed by a HashMap to sto elements. When an element is added:

1. It is hashed using the hashCode() method.
2. The hash code determines the bucket in the H where the element is stored.
3. If the bucket is empty, the element is added. I checked using equals(). If the element alread is not added again (ensuring uniqueness).

Q 8. What is the time complexity of operations in a H

Ans. The efficiency of hash table operations depends function and collision handling.

Operation	Average Case	Worst Case (when collisio
Insertion	O(1)	O(n)
Search	O(1)	O(n)

MULTI THREADING

Q 1. What is Multithreading, and why is it importan

Ans. Multithreading allows multiple tasks to execute within a program. In Java, it is achieved using t class or Runnable interface.

Importance of Multithreading in Java:

- Increased Performance: Utilizes CPU more e running multiple threads simultaneously.
- Improved Responsiveness: Helps keep appli responsive, especially in GUI-based progra
- Reduced Resource Usage: Threads share m reducing overhead compared to creating m processes.

Q 2. How can you create a thread in Java?

Ans. There are two ways to create a thread in Java:

1. Extending the Thread class
2. Implementing the Runnable interface

Q 3. What is the difference between a Process and

Ans. A process is an independent program in execut own memory space and system resources. Pro share memory and are heavier to create and m

MULTI THREADING

Q 4. How does synchronization work in Java? Explo concepts of synchronized methods and blocks

Ans. Synchronization in Java ensures that multiple t safely access shared resources without conflict race conditions, where multiple threads try to m same resource simultaneously.

There are two ways to implement synchronizati

- Synchronized Methods: A method declared synchronized keyword ensures that only one execute it at a time.
- Synchronized Blocks: Instead of synchronizin method, a synchronized block allows us to lo specific section of code using an object refe improves performance by reducing unnece locking.

Q 5. What is a deadlock, and how can it be avoided

Ans. A deadlock occurs when two or more threads c cycle, waiting for each other to release resourc preventing further execution.

To avoid deadlocks, we can:

MULTI THREADING

Q 6. What are the differences between the Thread c the Runnable interface in Java?

Ans. In Java, there are two ways to create a thread: the Thread class or by implementing the Runna interface.

Feature	Thread Class	Runnabl
Type	Concrete class	Abstrac
Inheritance	Extends the Thread class (cannot extend another class)	Can be implemen inte
Implementation	Must override run() method	Must impleme
Memory Usage	More memory required	Less mem
Flexibility	Less flexible (since class is already extending Thread)	More flexible inheritance

Using Runnable is generally preferred in larger as it promotes better separation of concerns a greater reusability.

Q 7. What is the purpose of the volatile keyword in

MULTI THREADING

Q 8. Explain the difference between preemptive s and time-slicing in the context of thread sch

Ans.

- Preemptive scheduling: The operating syst interrupt a running thread and assign CPU another thread based on priority. This ensu higher-priority tasks get executed first.
- Time-slicing: Each thread gets a fixed time run on the CPU. Once its time is up, the CPU to the next thread in a round-robin manne regardless of priority.

The key difference is that in preemptive sched thread can be interrupted anytime, whereas in slicing, a thread runs until its allocated time ex

EXCEPTION HANDLI

Q 1. What is an exception in Java, and why is excep handling important?

Ans. An exception in Java is an event that disrupts th flow of a program during execution. It is an obje thrown at runtime when an error occurs.

Importance of exception handling:

- Prevents program crashes.
- Helps recover from errors.
- Provides useful error information.
- Improves code robustness.
- Makes code easier to read and maintain.

Q 2. How does Java handle exceptions?

Ans. Java handles exceptions using exception propa When an exception is thrown, it moves up the co until it is caught by a matching catch block. If no found, the program terminates abnormally.

```
public class Example {

    public static void main(String[] args) {
        try {
            doSomethingThatMightThrowAnException();
        } catch (Exception e) {
            // Handle the exception here.
        }
    }
```

EXCEPTION HANDLI

Q 3. Describe the try-catch-finally block and its pu exception handling.

Ans. The try-catch-finally block in Java is used for ha exceptions gracefully. It consists of three parts:

- try block – Contains code that might throw a exception.
- catch block – Handles the exception if it occ
- finally block – Contains code that always ex regardless of whether an exception occurs o

Benefits of try-catch-finally:

- Prevents program crashes.
- Allows recovery from errors.
- Provides useful error information.
- Improves code robustness.
- Enhances code readability and maintainabi

Q 4. What is the difference between the throw and t keywords in Java?

Ans. The throw and throws keywords are used for ha exceptions in Java but serve different purposes.

- throw – Used inside a method to explicitly th

EXCEPTION HANDLI

Q 5. How can you create custom exceptions in Jav

Ans. To create a custom exception in Java, you nee
class that extends the Exception class (for chec
exceptions) or the RuntimeException class (for u
exceptions). The custom exception class can in
own constructors, methods, and fields to provid
details about the error.

```
public class MyException extends Exception {
    private String message;
    public MyException(String message) {
        super(message);
        this.message = message;
    }
    public String getMessage() {
        return message;
    }
```

ABOUT US

SkillCortex AI is an e-learning platform dedicated to pro
scenario-based teaching, ensuring students gain pract
industry-relevant experience. Our approach focuses on
learning without false promises, offering a structured sy
basic to advanced levels, along with 100% placement as

We provide high-standard materials, worksheets, assig
tests, and hackathons to enhance learning outcomes. C
benefit from a well-rounded education that prepares th
world challenges.

Every 40 days, we host free webinars featuring industry
keeping learners updated with the latest trends and ins
SkillCortex AI is committed to delivering quality educatio
the gap between learning and career success.

USP OF OUR PROGRAMS

- Real life scenario based high quality teach
- No fake promises of false internships and

SkillCortex AI

Redefining Education Through Innovation

FULL STACK COUR

Do you want to crack a high paying job but don't
Register in SkillCortex AI's 7 months upskilling pro
unlock your potential.

OUR HIGHLIGHTS

- Real life scenario based high quality te
- No fake promises of false internships ar
- 100% Placement Assistance.
- Syllabus designed from basic to advan
- Assignments, Mock tests, Hackathons a more.
- High Standard Materials and Workshee
- Free Webinars From Industry Experts Ev

Any Full Stack Course At Just