 What is inheritance in Java?

- A) A way to achieve code reusability
- B) A method for defining classes
- C) A process of creating an object
- D) A way to create interfaces

Answer: A

 Which keyword is used to inherit a class in Java?

- A) extends
- B) implements
- C) inherits
- D) super

Answer: A

 What type of inheritance allows a class to inherit from multiple classes?

- A) Single inheritance
- B) Multilevel inheritance
- C) Hierarchical inheritance
- D) Multiple inheritance (not allowed in Java)

Answer: D

 Which class is the universal super class in Java?

- A) Base
- B) Parent
- C) Object
- D) Super

Answer: C

 What is the purpose of the final keyword in class inheritance?

- A) To allow method overriding
- B) To prevent inheritance of the class
- C) To enable multiple inheritance
- D) To finalize a method

Answer: B

 Which access modifier allows the most restrictive access in inheritance?

- A) public
- B) private
- C) protected
- D) default

Answer: B

 In multilevel inheritance, which class is at the top of the hierarchy?

- A) The subclass
- B) The superclass
- C) The base class
- D) The derived class

Answer: B

 What does the keyword super refer to?

- A) The current class
- B) The parent class
- C) The object class
- D) The sibling class

Answer: B

 How can a subclass call a constructor of its superclass?

- A) By using this()
- B) By using super()
- C) By using call()
- D) By using parent()

Answer: B

 What happens when a method is overridden in a subclass?

- A) The parent class method is hidden
- B) The subclass method is called instead of the parent method

- C) Both methods are executed
- D) It causes a compile-time error

Answer: B

 What is dynamic method dispatch in Java?

- A) The process of determining which method to call at compile time
- B) The process of determining which method to call at runtime
- C) The process of defining abstract methods
- D) The process of overloading methods

Answer: B

 What is an abstract class?

- A) A class that cannot be instantiated
- B) A class with no methods
- C) A class that implements all methods
- D) A class that can have only static methods

Answer: A

 Which keyword is used to implement an interface in Java?

- A) extends
- B) implements
- C) inherits
- D) super

Answer: B

 What is the main advantage of using interfaces?

- A) To prevent method overriding
- B) To provide multiple inheritance
- C) To enforce method implementation
- D) To hide implementation details

Answer: C

 Which of the following allows a class to inherit from multiple interfaces?

- A) Single inheritance
- B) Multiple inheritance
- C) Interface inheritance
- D) None of the above

Answer: B

 What will happen if a subclass does not override a method from its superclass?

- A) It will throw an error
- B) It will inherit the superclass method
- C) It will create a new method
- D) It will be abstract

Answer: B

 What is the effect of declaring a method as final in a superclass?

- A) It can be overridden
- B) It cannot be overridden
- C) It is abstract
- D) It is static

Answer: B

 How can you define a method that must be implemented in any subclass?

- A) By declaring it as static
- B) By declaring it as final
- C) By declaring it as abstract
- D) By declaring it as private

Answer: C

 What is the result of trying to instantiate an abstract class?

- A) It will create an instance
- B) It will throw a compile-time error
- C) It will throw a runtime error
- D) It will create an empty object

Answer: B

 Which of the following is NOT a characteristic of interfaces?

- A) They can contain abstract methods
- B) They can contain concrete methods (default methods)
- C) They can extend other interfaces
- D) They can have instance variables

Answer: D

 What is an interface in Java?

- A) A class with no methods
- B) A reference type that can contain only constants, method signatures, default methods, static methods, and nested types
- C) A collection of classes
- D) A type of data structure

Answer: B

 How do you declare an interface in Java?

- A) class MyInterface {}
- B) interface MyInterface {}
- C) MyInterface {}
- D) declare interface MyInterface {}

Answer: B

 What must a class do to implement an interface?

- A) Use the extends keyword
- B) Use the implements keyword
- C) Use the interface keyword
- D) Use the inherit keyword

Answer: B

 Can a class implement multiple interfaces in Java?

- A) Yes
- B) No
- C) Only if the interfaces are related

- D) Only if the class is abstract

Answer: A

 Which of the following is true about nested interfaces?

- A) They cannot be static
- B) They are declared within another interface or class
- C) They cannot extend other interfaces
- D) They must be public

Answer: B

 How do interfaces support multiple inheritance?

- A) By allowing a class to extend multiple classes
- B) By allowing a class to implement multiple interfaces
- C) By using abstract classes
- D) By using default methods

Answer: B

 What is a default method in an interface?

- A) A method with no body
- B) A method that must be overridden
- C) A method with a default implementation
- D) A static method

Answer: C

 Which keyword is used to define a static method in an interface?

- A) static
- B) default
- C) abstract
- D) final

Answer: A

 What is a functional interface?

- A) An interface with more than one abstract method

- B) An interface with exactly one abstract method
- C) An interface that cannot be implemented
- D) An interface with only default methods

Answer: B

 Which of the following is a valid functional interface?

- A) interface MyFunctional { void method1(); void method2(); }
- B) interface MyFunctional { void method(); }
- C) interface MyFunctional { static void method(); }
- D) interface MyFunctional { default void method(); }

Answer: B

 How can you indicate that an interface is a functional interface?

- A) By using the @FunctionalInterface annotation
- B) By declaring it as final
- C) By implementing it as a class
- D) By using the static keyword

Answer: A

 What happens if a class implements multiple interfaces with the same method signature?

- A) Compilation error
- B) The class must provide an implementation
- C) It uses the implementation of the first interface
- D) It creates a new method

Answer: B

 Can an interface extend another interface?

- A) Yes, but only one
- B) Yes, multiple interfaces
- C) No, interfaces cannot extend
- D) Only if the parent interface is abstract

Answer: B

 Which of the following statements about annotations is true?

- A) Annotations provide metadata about a program
- B) Annotations can change the behavior of methods
- C) Annotations are a type of interface
- D) Annotations cannot be used on methods

Answer: A

 What is the purpose of the @Override annotation?

- A) To indicate a new method
- B) To signal that a method is being overridden from a superclass
- C) To declare an interface
- D) To define a default method

Answer: B

 How do you use the @FunctionalInterface annotation?

- A) Above an interface declaration to indicate it's a functional interface
- B) Inside a method to denote it as functional
- C) With classes only
- D) It is not a valid annotation

Answer: A

 Can an interface contain instance variables?

- A) Yes, but they must be static
- B) Yes, but they must be final
- C) No, interfaces cannot have instance variables
- D) Yes, but they cannot be public

Answer: C

 Which of the following is an example of a built-in functional interface in Java?

- A) Runnable
- B) Serializable
- C) Comparable
- D) Both A and C

Answer: D

 What is the difference between an abstract class and an interface?

- A) An abstract class can have constructors; an interface cannot
- B) An interface can have instance variables; an abstract class cannot
- C) An abstract class cannot have methods; an interface can
- D) There is no difference

Answer: A

 What is the purpose of default methods in interfaces?

- A) To allow backward compatibility
- B) To prevent method overriding
- C) To define abstract methods
- D) To declare static methods

Answer: A

 What is an array in Java?

- A) A single variable that can hold multiple values
- B) A data structure that holds values of different types
- C) A collection of variables of the same type
- D) A class that allows dynamic memory allocation

Answer: C

 How do you declare an array of integers in Java?

- A) int[] arr;
- B) int arr[];
- C) int arr;
- D) Both A and B

Answer: D

 Which of the following is the correct way to initialize an array?

- A) int[] arr = new int[5];

- B) int arr[] = {1, 2, 3, 4};
- C) int[] arr = new int[]{1, 2, 3, 4};
- D) All of the above

Answer: D

 Where are arrays stored in memory?

- A) Stack
- B) Heap
- C) Both stack and heap
- D) Registers

Answer: B

 How do you access the third element of an array named arr?

- A) arr[2]
- B) arr[3]
- C) arr[1]
- D) arr(2)

Answer: A

 What will be the output of the following code?

java
Copy code
int[] arr = {1, 2, 3, 4, 5};
System.out.println(arr[4]);

- A) 4
- B) 5
- C) 3
- D) Compilation error

Answer: B

 What happens if you try to access an index that is out of bounds in an array?

- A) The program continues running
- B) It throws an ArrayIndexOutOfBoundsException

- C) It returns null
- D) It crashes the JVM

Answer: B

 How do you assign one array to another?

- A) arr2 = arr1;
- B) arr2.copy(arr1);
- C) arr2 = arr1.clone();
- D) Both A and C

Answer: D

 Which of the following methods can be used to sort an array?

- A) Arrays.sort(arr);
- B) sort(arr);
- C) arr.sort();
- D) Arrays.order(arr);

Answer: A

 What is the time complexity of searching for a value in an unsorted array?

- A) O(log n)
- B) O(n)
- C) O(n log n)
- D) O(1)

Answer: B

 What is a two-dimensional array in Java?

- A) An array of arrays
- B) A single-dimensional array
- C) A method of storing objects
- D) None of the above

Answer: A

 How do you declare a two-dimensional array in Java?

- A) int[][] arr;
- B) int arr[][];
- C) int arr[2][3];
- D) Both A and B

Answer: D

 How do you initialize a three-dimensional array?

- A) int[][][] arr = new int[2][3][4];
- B) int arr[][][] = {{}};
- C) int[2][3][4] arr;
- D) Both A and B

Answer: D

 What is an array of varying lengths in Java called?

- A) Jagged array
- B) Multi-dimensional array
- C) Dynamic array
- D) Sparse array

Answer: A

 Which of the following is true about Java arrays?

- A) They can hold different data types.
- B) They have a fixed size once created.
- C) They cannot be resized dynamically.
- D) Both B and C

Answer: D

 Which class provides utility methods for arrays in Java?

- A) ArrayUtils
- B) Arrays
- C) ArrayHelper
- D) ArrayManager

Answer: B

 How do you find the length of an array named arr?

- A) arr.size()
- B) arr.length
- C) length(arr)
- D) arr.length()

Answer: B

 What is the default value of an integer array in Java?

- A) 0
- B) -1
- C) null
- D) Undefined

Answer: A

 What does the following code do?

java
Copy code

```
int[] arr = {1, 2, 3, 4, 5};
int sum = 0;
for (int i : arr) {
   sum += i;
}
```

- A) Sums all elements of the array
- B) Prints the array
- C) Initializes a new array
- D) Finds the maximum value

Answer: A

 Which of the following is true about arrays as vectors in Java?

- A) They can grow and shrink in size.
- B) They are fixed in size.
- C) They can hold different data types.
- D) They are not part of Java Collections Framework.

Answer: B

 What is the purpose of a package in Java?

- A) To define a variable
- B) To group related classes and interfaces
- C) To improve performance
- D) To create a new data type

Answer: B

 How do you define a package in a Java program?

- A) package name;
- B) define package name;
- C) create package name;
- D) import package name;

Answer: A

 Which of the following is the correct syntax to import a specific class from a package?

- A) import package.ClassName;
- B) import ClassName from package;
- C) using package.ClassName;
- D) include package.ClassName;

Answer: A

 What is the default package in Java?

- A) java
- B) default
- C) unnamed
- D) None of the above

Answer: C

 Which package is automatically imported by every Java program?

- A) java.util

- B) java.io
- C) java.lang
- D) java.awt

Answer: C

 What is the purpose of the ClassLoader in Java?

- A) To execute Java programs
- B) To load classes into the JVM
- C) To manage memory
- D) To compile Java code

Answer: B

 Which of the following classes is used for generating random numbers in Java?

- A) RandomNumber
- B) Math
- C) Random
- D) NumberGenerator

Answer: C

 What is auto-boxing in Java?

- A) Converting an object to a primitive type
- B) Converting a primitive type to an object
- C) Boxing multiple values
- D) None of the above

Answer: B

 What does the java.lang.Math class provide?

- A) File operations
- B) Mathematical functions
- C) String manipulation
- D) Networking functions

Answer: B

 Which class is used to format numbers and text in Java?

- A) Formatter
- B) FormatterClass
- C) Format
- D) StringFormatter

Answer: A

 What does the java.time.Instant class represent?

- A) A point in time
- B) A duration
- C) A date
- D) A time zone

Answer: A

 Which of the following is true about wrapper classes in Java?

- A) They convert primitive types to objects.
- B) They are used only for primitive types.
- C) They cannot be used with generics.
- D) They replace primitive types.

Answer: A

 What is the purpose of exception handling in programming?

- A) To improve performance
- B) To manage errors gracefully
- C) To increase code complexity
- D) To reduce code size

Answer: B

 Which keyword is used to explicitly throw an exception in Java?

- A) throw
- B) throws

- C) try
- D) catch

Answer: A

 What is the base class of all exceptions in Java?

- A) Exception
- B) Throwable
- C) Error
- D) RuntimeException

Answer: B

 Which of the following is an unchecked exception?

- A) IOException
- B) SQLException
- C) NullPointerException
- D) ClassNotFoundException

Answer: C

 What does the throws keyword indicate in a method signature?

- A) The method can handle the exception
- B) The method does not throw exceptions
- C) The method can throw certain exceptions
- D) The method must catch exceptions

Answer: C

 In which block can you handle exceptions?

- A) throw
- B) catch
- C) finally
- D) Both B and C

Answer: D

 What will happen if an exception is not caught?

- A) The program will continue normally
- B) The program will terminate
- C) The exception will be logged
- D) None of the above

Answer: B

 Which of the following is true about the finally block?

- A) It executes only if an exception is thrown
- B) It executes only if no exception is thrown
- C) It always executes, regardless of exceptions
- D) It can be skipped

Answer: C

 What is the hierarchy of standard exception classes in Java?

- A) Throwable > Error > Exception
- B) Exception > Throwable > Error
- C) Throwable > Exception > Error
- D) Exception > Error > Throwable

Answer: A

 Which statement about multiple catch clauses is true?

- A) You can only have one catch block per try block
- B) Catch blocks must be in the order of parent to child
- C) Multiple catch clauses cannot handle different exception types
- D) Only checked exceptions can be caught

Answer: B

 What type of exceptions are subclasses of RuntimeException?

- A) Checked exceptions
- B) Unchecked exceptions
- C) Error types
- D) All exceptions

Answer: B

 Which of the following exceptions must be declared in a method's throws clause?

- A) NullPointerException
- B) ClassCastException
- C) IOException
- D) ArithmeticException

Answer: C

 When is the catch block executed?

- A) Always, regardless of exceptions
- B) When no exceptions occur
- C) When a specific exception is thrown
- D) Before the try block

Answer: C

 What is the output if an exception is caught but not handled in the catch block?

- A) The program terminates normally
- B) The exception is ignored
- C) The exception is rethrown
- D) The program continues without any issues

Answer: C

 Which keyword is used to define a block of code to be tested for exceptions?

- A) throw
- B) throws
- C) try
- D) catch

Answer: C

 What will be the result of the following code?

java
Copy code

```
try {
    int a = 1 / 0;
```

```
} catch (ArithmeticException e) {
    System.out.println("Caught!");
} finally {
    System.out.println("Finally block executed");
}
```

- A) Caught!
- B) Finally block executed
- C) Both A and B
- D) None of the above

Answer: C

 Which type of exception indicates a serious problem that a typical application should not try to catch?

- A) RuntimeException
- B) Checked exceptions
- C) Error
- D) Exception

Answer: C

 What does the try block do in exception handling?

- A) It declares exceptions
- B) It handles exceptions
- C) It contains code that may throw exceptions
- D) It terminates the program

Answer: C

 Which of the following is a valid use of the finally block?

- A) To log exception details
- B) To close resources
- C) To perform cleanup operations
- D) All of the above

Answer: D

 Can a try block exist without a catch block?

- A) Yes, if a finally block is present
- B) No, a try block must have a catch
- C) Yes, but it must be empty
- D) No, it cannot exist at all

Answer: A

 Which package is primarily used for Java I/O operations?

- A) java.net
- B) java.util
- C) java.io
- D) java.nio

Answer: C) java.io

 Which of the following is a byte stream class in Java?

- A) FileReader
- B) BufferedReader
- C) FileInputStream
- D) PrintWriter

Answer: C) FileInputStream

 Which class is used to read character files in Java?

- A) FileOutputStream
- B) FileInputStream
- C) FileReader
- D) DataOutputStream

Answer: C) FileReader

 Which of the following streams is used to read and write data in a binary format?

- A) Character streams
- B) Byte streams
- C) Buffered streams
- D) Print streams

Answer: B) Byte streams

 What is the purpose of the Scanner class in Java?

- A) To perform arithmetic operations
- B) To parse and read input
- C) To write files
- D) To create GUI applications

Answer: B) To parse and read input

 Which method is used to read the next line of input using the Scanner class?

- A) readLine()
- B) nextLine()
- C) getLine()
- D) readNext()

Answer: B) nextLine()

 Which of the following is NOT a valid character stream class?

- A) FileReader
- B) FileWriter
- C) BufferedReader
- D) DataInputStream

Answer: D) DataInputStream

 What does the method Files.readAllLines(Path path) return?

- A) A single line from a file
- B) A byte array of the file contents
- C) A list of strings representing the lines in the file
- D) An InputStream of the file

Answer: C) A list of strings representing the lines in the file

 Which exception is thrown when a file is not found?

- A) IOException
- B) FileNotFoundException

- C) EOFException
- D) NullPointerException

Answer: B) FileNotFoundException

 Which class is used to write data to a file in Java?

- A) FileInputStream
- B) FileWriter
- C) PrintStream
- D) Both B and C

Answer: D) Both B and C

 Which of the following methods is used to close a stream in Java?

- A) closeStream()
- B) end()
- C) close()
- D) terminate()

Answer: C) close()

 Which method of the PrintWriter class is used to write a string to a file?

- A) writeString()
- B) print()
- C) write()
- D) Both B and C

Answer: D) Both B and C

 What is the base class for all Java I/O classes?

- A) Reader
- B) Writer
- C) InputStream
- D) Object

Answer: D) Object

 Which of the following classes provides a buffer for input and output in Java?

- A) FileReader
- B) BufferedReader
- C) FileOutputStream
- D) All of the above

Answer: B) BufferedReader

 Which I/O operation is performed using the Files class in Java?

- A) Writing data
- B) Reading data
- C) Deleting files
- D) All of the above

Answer: D) All of the above

 How do you create a new file using Java's I/O classes?

- A) new File("filename.txt");
- B) File.create("filename.txt");
- C) File.new("filename.txt");
- D) File.createNewFile("filename.txt");

Answer: D) File.createNewFile("filename.txt");

 What is the default character encoding used by FileReader in Java?

- A) UTF-8
- B) ISO-8859-1
- C) UTF-16
- D) System's default encoding

Answer: D) System's default encoding

 Which method of File class checks if a file exists?

- A) exists()
- B) isExist()
- C) checkExists()
- D) fileExists()

Answer: A) exists()

 What is the output of the following code?

java
Copy code
Scanner scanner = new Scanner(System.in);
System.out.println(scanner.next());

- A) Reads a full line
- B) Reads the next token
- C) Throws an exception
- D) Waits for user input indefinitely

Answer: B) Reads the next token

 Which of the following is a valid way to read from a file using BufferedReader?

- A) BufferedReader br = new BufferedReader(new FileReader("file.txt"));
- B) BufferedReader br = new BufferedReader("file.txt");
- C) BufferedReader br = new BufferedReader(new FileInputStream("file.txt"));
- D) Both A and C

Answer: D) Both A and C

 What is the primary class used for handling strings in Java?

- A) StringBuilder
- B) String
- C) StringBuffer
- D) CharSequence

Answer: B

 Which interface does the String class implement?

- A) Comparable
- B) CharSequence
- C) Serializable
- D) Cloneable

Answer: B

 What will the following code output?

java

Copy code

String str = "Hello";

System.out.println(str.charAt(1));

- A) H
- B) e
- C) l
- D) o

Answer: B

Which method is used to compare two strings in Java?

- A) equals()
- B) compare()
- C) isEqual()
- D) same()

Answer: A

How do you extract a substring from a string in Java?

- A) substring(startIndex, endIndex)
- B) extract(startIndex, endIndex)
- C) sub(startIndex, endIndex)
- D) slice(startIndex, endIndex)

Answer: A

What does the String.length() method return?

- A) The number of characters in the string
- B) The number of bytes in the string
- C) The memory size of the string
- D) The number of words in the string

Answer: A

Which method is used to convert a string to uppercase?

- A) toUpper()

- B) toUpperCase()
- C) upperCase()
- D) makeUppercase()

Answer: B

 Which class is used for mutable strings in Java?

- A) String
- B) CharSequence
- C) StringBuffer
- D) StringBuilder

Answer: C

 What will the following code output?

java
Copy code
String str = "Java";
str += " Programming";
System.out.println(str);

- A) Java
- B) Java Programming
- C) Java+Programming
- D) JavaProgramming

Answer: B

 Which method checks if a string starts with a specific prefix?

- A) startsWith()
- B) beginsWith()
- C) hasPrefix()
- D) isPrefix()

Answer: A

 What does the String.trim() method do?

- A) Removes whitespace from both ends of the string

- B) Shortens the string to a specified length
- C) Converts the string to lowercase
- D) Removes all spaces in the string

Answer: A

 How can you search for a character in a string?

- A) findChar()
- B) indexOf()
- C) searchChar()
- D) charIndex()

Answer: B

 Which of the following methods can be used to replace a character in a string?

- A) replace()
- B) change()
- C) modify()
- D) swap()

Answer: A

 What is the primary difference between StringBuffer and StringBuilder?

- A) StringBuffer is faster than StringBuilder
- B) StringBuilder is synchronized, StringBuffer is not
- C) StringBuffer is synchronized, StringBuilder is not
- D) There is no difference

Answer: C

 Which method would you use to concatenate two strings efficiently?

- A) String.concat()
- B) StringBuffer.append()
- C) StringBuilder.join()
- D) Both B and C

Answer: D

 What does the method String.split() do?

- A) Combines two strings
- B) Divides a string into an array of substrings
- C) Reverses the string
- D) Replaces a substring with another

Answer: B

 What will the following code output?

java
Copy code
String str = "Hello, World!";
System.out.println(str.contains("World"));

- A) true
- B) false
- C) Hello
- D) World

Answer: A

 Which method can be used to compare strings while ignoring case?

- A) equalsIgnoreCase()
- B) compareToIgnoreCase()
- C) equalsCase()
- D) caseInsensitiveEquals()

Answer: A

 What will happen if you try to modify a String object?

- A) It will throw an error
- B) It will modify the original string
- C) It will create a new string
- D) None of the above

Answer: C

 Which of the following is NOT a method of the String class?

- A) indexOf()
- B) lastIndexOf()
- C) reverse()
- D) substring()

Answer: C

 What is the primary benefit of multithreading in programming?

- A) Improved code readability
- B) Efficient CPU usage
- C) Simplified debugging
- D) Increased memory usage

Answer: B

 Which class is used to create a thread in Java?

- A) Runnable
- B) Thread
- C) Executor
- D) Task

Answer: B

 What is the state of a thread when it is waiting for a resource?

- A) Running
- B) Blocked
- C) New
- D) Terminated

Answer: B

 Which method is used to start a thread in Java?

- A) run()
- B) start()
- C) execute()
- D) begin()

Answer: B

 What is the default priority of a thread in Java?

- A) 1
- B) 5
- C) 10
- D) 0

Answer: B

 What does the join() method do in thread programming?

- A) Combines two threads
- B) Waits for a thread to die
- C) Starts a thread
- D) Resumes a suspended thread

Answer: B

 Which of the following methods is used to pause a thread?

- A) sleep()
- B) wait()
- C) suspend()
- D) halt()

Answer: A

 What is a deadlock in multithreading?

- A) A situation where all threads are running
- B) A situation where two or more threads are blocked forever
- C) A thread that is terminated
- D) A thread that is paused

Answer: B

 Which of the following can cause a race condition?

- A) Synchronization
- B) Multiple threads accessing shared data
- C) Using locks
- D) Proper thread management

Answer: B

 What is the purpose of thread synchronization?

- A) To reduce memory usage
- B) To prevent data inconsistency
- C) To increase thread priority
- D) To speed up execution

Answer: B

 Which keyword is used to implement synchronization in Java?

- A) synchronized
- B) lock
- C) mutex
- D) monitor

Answer: A

 What is the role of the wait() method in inter-thread communication?

- A) It starts a new thread
- B) It releases the lock on an object
- C) It pauses the current thread indefinitely
- D) It terminates a thread

Answer: B

 What does the notify() method do in Java?

- A) It starts a new thread
- B) It wakes up a single thread waiting on the object's monitor
- C) It stops a thread
- D) It blocks the current thread

Answer: B

 Which of the following is a valid thread state?

- A) Running
- B) Sleeping

- C) Blocked
- D) All of the above

Answer: D

 What is the effect of calling Thread.sleep()?

- A) It terminates the thread
- B) It pauses the thread for a specified time
- C) It increases the thread's priority
- D) It creates a new thread

Answer: B

 What does the Thread.yield() method do?

- A) It stops the current thread
- B) It releases the current thread's resources
- C) It suggests to the thread scheduler to pause the current thread
- D) It terminates the thread

Answer: C

 How can you avoid a deadlock situation?

- A) Use more threads
- B) Limit resource allocation
- C) Use synchronized blocks
- D) Implement a timeout mechanism

Answer: D

 What happens when a thread is in the "new" state?

- A) The thread is ready to run
- B) The thread is currently running
- C) The thread has completed execution
- D) The thread has been created but not yet started

Answer: D

 Which method would you use to stop a thread in Java?

- A) stop()
- B) terminate()
- C) interrupt()
- D) exit()

Answer: C

 Which class provides thread pooling in Java?

- A) ThreadPool
- B) ExecutorService
- C) ThreadGroup
- D) Runnable

Answer: B

 What is JavaFX primarily used for?

- A) Networking
- B) Database management
- C) Creating rich desktop applications
- D) Web development

Answer: C

 What tool is commonly used to design JavaFX user interfaces visually?

- A) JavaFX Builder
- B) JavaFX Scene Builder
- C) JavaFX Designer
- D) JavaFX Layout Tool

Answer: B

 Which of the following is the top-level container for a JavaFX application?

- A) Scene
- B) Stage
- C) Node
- D) Application

Answer: B

 In JavaFX, what class is used to create a scene?

- A) SceneBuilder
- B) Scene
- C) Stage
- D) Application

Answer: B

 Which method is called to display the primary stage in a JavaFX application?

- A) show()
- B) display()
- C) render()
- D) start()

Answer: A

 How can you display text in a JavaFX application?

- A) TextField
- B) TextArea
- C) Label
- D) Text

Answer: D

 Which class is used to display an image in JavaFX?

- A) ImageView
- B) ImageDisplay
- C) Picture
- D) PhotoView

Answer: A

 What is the purpose of the HBox layout in JavaFX?

- A) To arrange nodes vertically
- B) To arrange nodes in a grid
- C) To arrange nodes horizontally
- D) To overlap nodes

Answer: C

 How do you handle mouse events in JavaFX?

- A) By using MouseEvent class
- B) By implementing MouseListener
- C) By using event handlers
- D) By using MouseHandler interface

Answer: C

 Which method is used to add an event handler to a button in JavaFX?

- A) setOnClick()
- B) setOnAction()
- C) addActionListener()
- D) registerAction()

Answer: B

 What does the setAlignment method do in a layout container?

- A) Sets the size of the container
- B) Sets the alignment of the child nodes
- C) Sets the background color of the container
- D) Sets the visibility of the container

Answer: B

 Which layout allows for a flexible arrangement of nodes in JavaFX?

- A) FlowPane
- B) BorderPane
- C) GridPane
- D) StackPane

Answer: A

 Which method is called to initialize the JavaFX application?

- A) init()
- B) start()

- C) launch()
- D) configure()

Answer: B

 What is a Scene Graph in JavaFX?

- A) A representation of a user interface
- B) A database structure
- C) A method for event handling
- D) A type of image rendering

Answer: A

 Which event is triggered when a mouse button is pressed and released?

- A) MouseClicked
- B) MousePressed
- C) MouseReleased
- D) MouseEntered

Answer: A

 How can you change the background color of a JavaFX scene?

- A) Using setBackground()
- B) Using setStyle() with CSS
- C) Using setColor()
- D) It is not possible

Answer: B

 What class is used to create a button in JavaFX?

- A) Button
- B) PushButton
- C) ClickableButton
- D) ActionButton

Answer: A

 Which of the following is true about JavaFX properties?

- A) They are immutable
- B) They can be bound to other properties
- C) They do not support change listeners
- D) They are only for UI components

Answer: B

 How do you remove a node from a layout in JavaFX?

- A) remove(node)
- B) delete(node)
- C) detach(node)
- D) clear(node)

Answer: A

 What is the purpose of the EventHandler interface in JavaFX?

- A) To define custom graphics
- B) To handle events generated by UI components
- C) To manage application states
- D) To create new UI components

Answer: B