Technical

1. What is the output of the following C code?

```
#include <stdio.h>
int main() {
   int i = 0;
   for (i = 0; i < 5; i++);
   printf("%d", i);
   return 0;
}
```

A) 4

B) 5

C) 0

D) Compilation Error

Answer: B) 5

Explanation: The semicolon after the for loop `for (i = 0; i < 5; i++);` makes the loop an empty loop. The loop runs 5 times, incrementing `i` from 0 to 4. On the last check, `i` becomes 5, the condition `5 < 5` is false, and the loop terminates. The following `printf` statement then prints the final value of `i`, which is 5.

2. What is the output of the following C++ code?

```
#include <iostream>
class Test {
public:
   Test() { std::cout << "Constructor Called\n"; }
   ~Test() { std::cout << "Destructor Called\n"; }
};
int main() {
   Test *t = (Test*)malloc(sizeof(Test));
   free(t);
```

```
    return 0;
}
```

A) Constructor Called

  Destructor Called

B) Constructor Called

C) Destructor Called

D) No output

Answer: D) No output

Explanation: `malloc` only allocates raw memory; it does not call the constructor of the class. Similarly, `free` only deallocates the memory; it does not call the destructor. To properly construct and destruct the object, `new` and `delete` should be used.

3. What is the result of the following Java code snippet?

```
public class Main {
    public static void main(String[] args) {
        String s1 = "Java";
        String s2 = "Java";
        s1.replace('J', 'L');
        System.out.println(s1 == s2);
    }
}
```

A) true

B) false

C) Compilation Error

D) Throws Exception

Answer: A) true

Explanation: Both `s1` and `s2` refer to the same string literal "Java" in the string pool. The `replace()` method in Java returns a new string and does not modify the original string because strings are immutable. Since `s1` is not reassigned, it still points to the original "Java" object. Therefore, `s1 == s2` is true.

4. Which of the following SQL statements will result in an error?

A) SELECT * FROM Employees WHERE Salary > 50000;

B) SELECT Department, COUNT(*) FROM Employees GROUP BY Department;

C) SELECT Department, COUNT(*) FROM Employees WHERE COUNT(*) > 10 GROUP BY Department;

D) SELECT Department, COUNT(*) FROM Employees GROUP BY Department HAVING COUNT(*) > 10;

Answer: C) SELECT Department, COUNT(*) FROM Employees WHERE COUNT(*) > 10 GROUP BY Department;

Explanation: Aggregate functions like `COUNT(*)` cannot be used in a `WHERE` clause. The `WHERE` clause filters rows before any grouping occurs. To filter groups based on an aggregate function, the `HAVING` clause must be used after the `GROUP BY` clause.

5. What is the time complexity to find an element in a balanced Binary Search Tree (BST) with n nodes?

A) O(n)

B) O(1)

C) O(log n)

D) O(n^2)

Answer: C) O(log n)

Explanation: In a balanced Binary Search Tree, the height of the tree is O(log n). Since the search operation proceeds from the root to a leaf, the maximum number of comparisons is proportional to the height of the tree, resulting in a time complexity of O(log n).

6. Predict the output of this C program.

```
#include <stdio.h>
int main() {
    float f = 0.5;
    if (f == 0.5f) {
        printf("Equal");
    } else {
        printf("Not Equal");
    }
    return 0;
}
```

A) Equal

B) Not Equal

C) Undefined Behavior

D) Compilation Error

Answer: A) Equal

Explanation: By default, a floating-point literal like `0.5` is treated as a `double`. The variable `f` is a `float`. Comparing a `float` with a `double` can lead to precision issues. However, by using the suffix `f` (i.e., `0.5f`), we specify that the literal is a `float`, ensuring the comparison is between two values of the same type and precision, which results in "Equal".

7. What does the following C++ code print?

```
#include <iostream>
int main() {
    int arr[] = {10, 20, 30};
    int &ref = arr[0];
    ref++;
    std::cout << arr[0] << std::endl;
    return 0;
}
```

A) 10

B) 11

C) Garbage Value

D) Compilation Error

Answer: B) 11

Explanation: A reference `ref` is created for the first element of the array `arr[0]`. When `ref` is incremented, it modifies the value it refers to, which is `arr[0]`. So, `arr[0]` becomes 11.

8. What happens when this Java code is executed?

```
public class Main {
    public static void main(String[] args) {
        final StringBuilder sb = new StringBuilder("Hello");
        sb.append(" World");
        System.out.println(sb);
    }
}
```

A) Hello

B) Hello World

C) Compilation Error

D) Throws Exception

Answer: B) Hello World

Explanation: The `final` keyword, when applied to an object reference like `sb`, means that the reference variable cannot be reassigned to point to another object. However, the state of the object itself can be modified. `StringBuilder` is mutable, so `append(" World")` modifies the object's content.

9. What will be the result of this SQL query?

```
SELECT CASE WHEN NULL = NULL THEN 'True' ELSE 'False' END;
```

A) True

B) False

C) NULL

D) Error

Answer: B) False

Explanation: In SQL, comparing `NULL` with `NULL` using the `=` operator results in `UNKNOWN`, which is treated as `false` in a `CASE` statement's `WHEN` condition. To check for `NULL`, you must use the `IS NULL` operator.

10. Which data structure is most suitable for implementing a "Undo" feature in a text editor?

A) Queue

B) Stack

C) Linked List

D) Tree

Answer: B) Stack

Explanation: An "Undo" feature requires actions to be reversed in the opposite order they were performed. A Stack follows the Last-In, First-Out (LIFO) principle, which is the exact behavior needed. The most recent action is pushed onto the stack, and "Undo" pops it off.

11. What is the value of `x` after this C code runs?

```
#include <stdio.h>
int main() {
   int x = 10;
   x = x << 2;
   printf("%d", x);
   return 0;
}
```

A) 20

B) 40

C) 5

D) 2

Answer: B) 40

Explanation: The `<<` is the left bitwise shift operator. `x << 2` shifts the bits of `x` two places to the left. The binary representation of 10 is `00001010`. Shifting left by 2 results in `00101000`, which is the decimal number 40. Left shifting by `n` is equivalent to multiplying by `2^n`.

12. What is the output of the C++ code below?

```
#include <iostream>
class Base {
public:
   virtual void show() { std::cout << "Base\n"; }
};
```

```
class Derived : public Base {
public:
    void show() { std::cout << "Derived\n"; }
};
int main() {
    Base b;
    Derived d;
    Base *ptr = &b;
    ptr->show();
    ptr = &d;
    ptr->show();
    return 0;
}
```

A) Base
   Base

B) Derived
   Derived

C) Base
   Derived

D) Derived
   Base

Answer: C) Base
   Derived

Explanation: This demonstrates runtime polymorphism. When `ptr` points to a `Base` object, `Base::show()` is called. When `ptr` points to a `Derived` object, the `virtual` keyword ensures that the overridden `Derived::show()` method is called through the base class pointer.

13. What is printed by this Java code?

```
import java.util.ArrayList;
import java.util.List;
public class Main {
    public static void main(String[] args) {
        List<String> list = new ArrayList<>();
        list.add("A");
        list.add(1, "B");
        list.set(0, "C");
        System.out.println(list);
    }
}
```

A) [A, B]

B) [C, B]

C) [C, A]

D) [A, C]

Answer: B) [C, B]

Explanation: `list.add("A")` adds "A" at index 0. The list is `[A]`. `list.add(1, "B")` adds "B" at index 1. The list becomes `[A, B]`. `list.set(0, "C")` replaces the element at index 0 with "C". The final list is `[C, B]`.

14. What does the following SQL query do?

```
DELETE FROM Products;
```

A) Deletes the `Products` table.

B) Deletes all rows from the `Products` table.

C) Deletes all columns from the `Products` table.

D) This query is invalid.

Answer: B) Deletes all rows from the `Products` table.

Explanation: The `DELETE FROM table_name` command without a `WHERE` clause removes all rows from the specified table. The table structure itself remains. This operation can typically be rolled back.

15. In a circular queue implemented with an array of size N, how is the front pointer advanced?

A) front = front + 1;

B) front = (front + 1) % N;

C) front = front - 1;

D) front = (front - 1 + N) % N;

Answer: B) front = (front + 1) % N;

Explanation: In a circular queue, the pointers wrap around to the beginning of the array when they reach the end. The modulo operator (`%`) is used to achieve this circular behavior. `(front + 1) % N` correctly advances the pointer and wraps it around if it reaches `N`.

16. What is the output of this C code involving the preprocessor?

```
#include <stdio.h>

#define MERGE(a, b) a##b

int main() {

   int xy = 100;

   printf("%d\n", MERGE(x, y));

   return 0;

}
```

A) xy

B) 100

C) 0

D) Compilation Error

Answer: B) 100

Explanation: The `##` operator in a C preprocessor macro is the token-pasting operator. It concatenates the two tokens `a` and `b`. So, `MERGE(x, y)` becomes `xy`. The program then prints the value of the integer variable `xy`, which is 100.

17. What is the concept demonstrated by the following C++ code?

```
#include <iostream>

class Point {

private:

   int x, y;
public:
   Point(int x1, int y1) { x = x1; y = y1; }
   Point operator+(const Point &p) {
       return Point(x + p.x, y + p.y);
   }
   void print() { std::cout << x << " " << y << std::endl; }
};
int main() {
   Point p1(10, 20), p2(5, 15);
   Point p3 = p1 + p2;
   p3.print();
   return 0;
}
```

A) Function Overloading

B) Operator Overloading

C) Function Overriding

D) Inheritance

Answer: B) Operator Overloading

Explanation: The code defines a custom behavior for the `+` operator for the `Point` class. This allows objects of the `Point` class to be added together using the `+` symbol, which is a classic example of operator overloading.

18. Predict the output of the Java code.

```
public class Main {
    public static void main(String[] args) {
        try {
            System.out.print("A");
            int value = 5 / 0;
            System.out.print("B");
        } catch (Exception e) {
            System.out.print("C");
        } finally {
            System.out.print("D");
        }
        System.out.print("E");
    }
}
```

A) ACDE

B) ABCDE

C) AD

D) ACD

Answer: A) ACDE

Explanation: "A" is printed. Then, `5 / 0` throws an `ArithmeticException`. The `catch` block catches this exception, and "C" is printed. The `finally` block is always executed, so "D" is printed. The program then continues execution after the try-catch-finally block, and "E" is printed.

19. Given a table `Students` with columns `StudentID` and `Score`, which query finds the second highest score?

A) SELECT MAX(Score) FROM Students WHERE Score < (SELECT MAX(Score) FROM Students);

B) SELECT Score FROM Students ORDER BY Score DESC LIMIT 1, 1;

C) SELECT MAX(Score) FROM Students WHERE Score NOT IN (SELECT MAX(Score) FROM Students);

D) All of the above could work depending on the SQL dialect.

Answer: D) All of the above could work depending on the SQL dialect.

Explanation: A uses a subquery to exclude the highest score and find the max of the rest. C is similar to A. B uses `LIMIT` with an offset, which works in MySQL and PostgreSQL but not in all SQL versions (e.g., SQL Server would use `OFFSET FETCH`). All three approaches are valid ways to find the second highest value.

20. What is a key difference between a graph and a tree?

A) Graphs can have cycles, while trees cannot.

B) Trees can have cycles, while graphs cannot.

C) Graphs are always connected, while trees are not.

D) Trees must have a root, while graphs do not.

Answer: A) Graphs can have cycles, while trees cannot.

Explanation: A tree is a specific type of connected, acyclic graph. The defining characteristic that separates a general graph from a tree is that a tree must not contain any cycles.

21. What is the output of this C code snippet?

```
#include <stdio.h>
int main() {
   const int val = 10;
   int *ptr = (int*) &val;
   *ptr = 20;
   printf("%d", val);
   return 0;
}
```

A) 10

B) 20

C) Compilation Error

D) Undefined Behavior

Answer: D) Undefined Behavior

Explanation: Attempting to modify a `const` qualified variable through a pointer results in undefined behavior. The compiler might place `val` in a read-only memory segment. While the code may compile (with a warning), its runtime behavior is not guaranteed by the C standard. It might print 10, 20, or crash.

22. What is a friend function in C++?

A) A function that can access private members of any class.

B) A member function of a class that is accessible from outside the class.

C) A non-member function that is granted access to the private and protected members of a class.

D) A function that is inherited by a derived class.

Answer: C) A non-member function that is granted access to the private and protected members of a class.

Explanation: A class can declare a function as its "friend". This gives the friend function access to the class's private and protected members, even though the function is not a member of the class itself.

23. What is the state of a thread when it is waiting for another thread to complete its task?

A) RUNNABLE

B) BLOCKED

C) WAITING

D) TIMED_WAITING

Answer: C) WAITING

Explanation: A thread enters the `WAITING` state when it calls a method like `Object.wait()` without a timeout or `Thread.join()` without a timeout. It is waiting indefinitely for another thread to perform a particular action, such as calling `notify()` or terminating.

24. What is the primary purpose of the SQL `UNION` operator?

A) To join columns from two different tables.

B) To combine the result sets of two or more SELECT statements.

C) To create a new table from existing tables.

D) To filter duplicate rows from a single query.

Answer: B) To combine the result sets of two or more SELECT statements.

Explanation: The `UNION` operator is used to append the results of two queries. It combines the rows from both queries into a single result set. By default, `UNION` removes duplicate rows from the combined result.

25. What is the worst-case time complexity for searching, inserting, and deleting in a general Binary Tree?

A) O(log n)

B) O(n)

C) O(1)

D) O(n log n)

Answer: B) O(n)

Explanation: In the worst case, a binary tree can be skewed, resembling a linked list. In such a scenario, the height of the tree is `n`, and operations that depend on the height, like search, insertion, and deletion, will have a time complexity of O(n).

26. What is printed by the following C program?

```
#include <stdio.h>
union Data {
   int i;
   char c;
};
int main() {
   union Data d;
   d.i = 65;
   printf("c = %c\n", d.c);
   return 0;
}
```

A) c = A

B) c = 65

C) c = (some garbage value)

D) Compilation Error

Answer: A) c = A

Explanation: In a union, all members share the same memory location. The integer value 65 is stored in `d.i`. The ASCII representation of 65 is the character 'A'. When `d.c` is accessed, the byte at that memory location is interpreted as a character, which is 'A'.

27. Which C++ feature allows a derived class to provide a specific implementation of a method that is already provided by its base class?

A) Abstraction

B) Function Overloading

C) Method Overriding

D) Encapsulation

Answer: C) Method Overriding

Explanation: Method overriding is a feature of inheritance where a subclass provides a different implementation for a method that is defined in its superclass. This is a key part of runtime polymorphism.

28. In Java, what is the role of the Garbage Collector?

A) To manually deallocate memory for objects.

B) To automatically manage the lifecycle of threads.

C) To automatically identify and reclaim memory from objects that are no longer reachable.

D) To collect and report runtime exceptions.

Answer: C) To automatically identify and reclaim memory from objects that are no longer reachable.

Explanation: The Garbage Collector (GC) is a form of automatic memory management. It runs in the background, tracking objects in the heap and freeing up memory occupied by objects that are no longer in use by the program.

29. What is the difference between a Primary Key and a Unique Key constraint in SQL?

A) A table can have multiple Primary Keys, but only one Unique Key.

B) A Primary Key cannot contain NULL values, whereas a Unique Key can.

C) A Unique Key cannot contain NULL values, whereas a Primary Key can.

D) There is no difference.

Answer: B) A Primary Key cannot contain NULL values, whereas a Unique Key can.

Explanation: Both constraints ensure that the values in a column (or a set of columns) are unique. However, a Primary Key is stricter: it does not allow NULL values, and a table can have only one Primary Key. A Unique Key constraint allows one NULL value (in most database systems).

30. Which sorting algorithm is considered stable?

A) Quicksort

B) Heapsort

C) Selection Sort

D) Merge Sort

Answer: D) Merge Sort

Explanation: A sorting algorithm is stable if it preserves the relative order of equal elements. Merge Sort is a stable sorting algorithm. Quicksort and Heapsort are not stable. Selection Sort is generally not stable.

31. What is a dangling pointer in C?

A) A pointer that has not been initialized.

B) A pointer that points to a memory location that has been deallocated (freed).

C) A pointer that is explicitly set to NULL.

D) A pointer to a pointer.

Answer: B) A pointer that points to a memory location that has been deallocated (freed).

Explanation: A dangling pointer arises when a pointer continues to refer to a memory address after the memory has been freed. Accessing a dangling pointer leads to undefined behavior.

32. What is the output of the following C++ program?

```
#include <iostream>
struct A {
  int i;
  A(int i) : i(i) {}
  ~A() { std::cout << "Destruct " << i << std::endl; }
};
int main() {
  A a1(1);
  A a2(2);
  return 0;
}
```

A) Destruct 1

 Destruct 2

B) Destruct 2

Destruct 1

C) No output

D) Undefined Behavior

Answer: B) Destruct 2

Destruct 1

Explanation: Objects created on the stack are destructed in the reverse order of their creation when they go out of scope. `a1` is created first, then `a2`. When `main` exits, `a2` is destructed first, followed by `a1`.

33. Which collection class in Java does not allow duplicate elements?

A) ArrayList

B) LinkedList

C) HashSet

D) HashMap

Answer: C) HashSet

Explanation: `HashSet` is an implementation of the `Set` interface. A key property of a Set is that it does not store duplicate elements. If you try to add an element that already exists in the `HashSet`, the add operation will be ignored.

34. What does `LEFT JOIN` do in SQL?

A) Returns all records from the right table, and the matched records from the left table.

B) Returns records that have matching values in both tables.

C) Returns all records from the left table, and the matched records from the right table.

D) Returns all records when there is a match in either left or right table.

Answer: C) Returns all records from the left table, and the matched records from the right table.

Explanation: A `LEFT JOIN` (or `LEFT OUTER JOIN`) returns all rows from the left table, along with the matching rows from the right table. If there is no match in the right table, the result is `NULL` for the columns from the right table.

35. What is the data structure used to manage function calls and recursion?

A) Queue

B) Heap

C) Stack

D) Array

Answer: C) Stack

Explanation: The call stack is a stack data structure that stores information about the active subroutines (functions) of a computer program. Each time a function is called, a new frame is pushed onto the stack, and when the function returns, its frame is popped off.

36. What will be the value of `c`?

```
#include <stdio.h>
int main() {
   int a = 5, b = 3;
   int c = a & b;
   printf("%d", c);
   return 0;
}
```

A) 1

B) 2

C) 5

D) 7

Answer: A) 1

Explanation: The `&` is the bitwise AND operator. The binary representation of 5 is `0101` and for 3 is `0011`. Performing a bitwise AND operation:

`0101 & 0011 = 0001`, which is the decimal number 1.

37. Which OOPS concept is best represented by a real-world example of a car, which hides its internal complexity (engine, transmission) from the driver?

A) Inheritance

B) Polymorphism

C) Encapsulation/Abstraction

D) Overloading

Answer: C) Encapsulation/Abstraction

Explanation: Encapsulation is the bundling of data with the methods that operate on that data. Abstraction is the concept of hiding the complex reality while exposing only the necessary parts. A driver interacts with a car through a simple interface (steering wheel, pedals) without needing to know the complex internal mechanics, which is a perfect example of abstraction.

38. What is the purpose of an assertion in Java?

A) To handle runtime errors gracefully.

B) To define a condition that must be true at a particular point in the code.

C) To replace a standard `if-else` statement.

D) To manage memory allocation.

Answer: B) To define a condition that must be true at a particular point in the code.

Explanation: Assertions are used for debugging purposes. An assertion is a statement that checks a condition which is expected to be true. If the condition is false, an `AssertionError` is thrown, which typically halts the program, indicating a bug. Assertions must be explicitly enabled at runtime.

39. What is the main difference between `TRUNCATE` and `DELETE` in SQL?

A) `TRUNCATE` is faster and uses fewer system resources than `DELETE`.

B) `TRUNCATE` is a DDL command, while `DELETE` is a DML command.

C) `TRUNCATE` resets identity columns, while `DELETE` does not.

D) All of the above.

Answer: D) All of the above.

Explanation: `TRUNCATE` is a Data Definition Language (DDL) command that quickly removes all rows from a table, often by deallocating the data pages. It cannot be rolled back easily, resets identity counters, and doesn't fire delete triggers. `DELETE` is a Data Manipulation Language (DML) command that removes rows one by one, can be rolled back, and fires triggers.

40. What is the height of a complete binary tree with `n` nodes?

A) O(n)

B) O(log n)

C) O(1)

D) O(n^2)

Answer: B) O(log n)

Explanation: A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes in the last level are as far left as possible. This structure ensures that the height of the tree is logarithmic with respect to the number of nodes, specifically `floor(log2(n))`.

41. What is the output of the following C code?

```
#include <stdio.h>

#include <string.h>

int main() {

  char str[10] = "Hello";

  printf("%zu %zu\n", sizeof(str), strlen(str));

  return 0;

}
```

A) 5 5

B) 10 5

C) 6 5

D) 10 6

Answer: B) 10 5

Explanation: `sizeof(str)` returns the total size of the array `str` in bytes, which was declared to be 10. `strlen(str)` returns the length of the C-string, which is the number of characters before the null terminator (`\0`). The string "Hello" has 5 characters.

42. In C++, if a class `D` inherits from class `B`, and an object of class `D` is created, which constructor is called first?

A) The constructor of the derived class `D`.

B) The constructor of the base class `B`.

C) It depends on the order of declaration.

D) Both are called simultaneously.

Answer: B) The constructor of the base class `B`.

Explanation: During the construction of a derived class object, the base class constructor is always called before the derived class constructor. This ensures that the base part of the object is properly initialized before the derived part is constructed.

43. What does `java.lang.Object` class represent in Java?

A) It is the root class for all custom exception types.

B) It is a utility class for mathematical operations.

C) It is the root of the class hierarchy; every class is a descendant of `Object`.

D) It is a class used to create instances of other classes.

Answer: C) It is the root of the class hierarchy; every class is a descendant of `Object`.

Explanation: In Java, every class, whether explicitly declared or not, inherits from the `java.lang.Object` class. This makes it the ultimate superclass of all classes and ensures that every object has a common set of methods like `equals()`, `hashCode()`, and `toString()`.

44. Which constraint ensures that all values in a column are different from each other and also prevents NULL values?

A) UNIQUE

B) NOT NULL

C) CHECK

D) PRIMARY KEY

Answer: D) PRIMARY KEY

Explanation: A `PRIMARY KEY` is a combination of a `UNIQUE` constraint and a `NOT NULL` constraint. It uniquely identifies each record in a table and does not allow any `NULL` values.

45. Which of the following data structures provides the fastest access to an element given its index?

A) Singly Linked List

B) Doubly Linked List

C) Array

D) Queue

Answer: C) Array

Explanation: An array stores elements in contiguous memory locations. This allows for direct calculation of an element's memory address based on its index, providing O(1) or constant time access. Linked lists require traversing from the head of the list, which takes O(n) time.

46. What does this C declaration mean: `int (*fp)(char*);`?

A) A function `fp` that takes a `char*` and returns an `int*`.

B) A pointer to a function that takes a `char*` and returns an `int`.

C) A function `fp` that takes an `int` and returns a `char*`.

D) An array of pointers to functions.

Answer: B) A pointer to a function that takes a `char*` and returns an `int`.

Explanation: The parentheses around `*fp` give the pointer operator `*` precedence over the function call operator `()`. This means `fp` is a pointer. It points to a function that takes a `char*` as an argument and returns an `int`.

47. In C++, what happens if a class has a pointer to dynamically allocated memory but does not have a user-defined copy constructor?

A) The code will not compile.

B) A shallow copy will be performed by the default copy constructor.

C) A deep copy will be performed by the default copy constructor.

D) A runtime error will always occur.

Answer: B) A shallow copy will be performed by the default copy constructor.

Explanation: If a copy constructor is not provided, the compiler generates a default one that performs a member-wise copy (a shallow copy). If the class contains a pointer, only the pointer's address is copied, not the data it points to. This leads to two objects pointing to the same memory, which can cause issues like double-free errors.

48. Which of these is a checked exception in Java?

A) `NullPointerException`

B) `ArrayIndexOutOfBoundsException`

C) `IOException`

D) `ArithmeticException`

Answer: C) `IOException`

Explanation: Checked exceptions are exceptions that are checked at compile-time. Methods that can throw a checked exception must either handle it using a try-catch block or declare it in the method signature using the `throws` keyword. `IOException` is a classic example. The others are unchecked (runtime) exceptions.

49. What is an SQL injection?

A) A database performance tuning technique.

B) A code injection technique that might destroy your database.

C) A method for adding new data to a table.

D) A type of database join.

Answer: B) A code injection technique that might destroy your database.

Explanation: SQL injection is a security vulnerability where an attacker can interfere with the queries that an application makes to its database. It allows an attacker to view data they are not normally able to retrieve or even modify or delete data, causing persistent changes to the application's content or behavior.

50. What is the main advantage of a doubly linked list over a singly linked list?

A) It uses less memory.

B) It can be traversed in both forward and reverse directions.

C) Insertion and deletion are faster.

D) It allows for direct access to any element.

Answer: B) It can be traversed in both forward and reverse directions.

Explanation: A doubly linked list contains an extra pointer in each node, known as the previous pointer, which points to the preceding node. This additional pointer allows for traversal in both directions, which can simplify certain operations like deletion of a specific node.

51. What is the purpose of the `typedef` keyword in C?

A) To create a new variable of a certain type.

B) To define a new data type from scratch.

C) To create an alias or a new name for an existing data type.

D) To check the type of a variable at runtime.

Answer: C) To create an alias or a new name for an existing data type.

Explanation: `typedef` is used to give a new name to an existing data type. This is often used to make complex type declarations more readable, for example, `typedef unsigned char BYTE;`.

52. Which statement is true about exception handling in C++?

A) Exceptions must be of primitive types like `int` or `char`.

B) A `try` block must be followed by exactly one `catch` block.

C) Any type of object, including user-defined classes, can be thrown as an exception.

D) The `finally` keyword is used to ensure code is always executed.

Answer: C) Any type of object, including user-defined classes, can be thrown as an exception.

Explanation: In C++, you can throw a value of any type as an exception, including integers, strings, or objects of user-defined classes. A `try` block can be followed by multiple `catch` blocks to handle different exception types. C++ does not have a `finally` keyword like Java; resource management is typically handled using RAII (Resource Acquisition Is Initialization).

53. What will be the output of this Java program?

```
public class Main {
  public static void main(String args[]) {
    System.out.println(10 + 20 + "Java");
    System.out.println("Java" + 10 + 20);
  }
}
```

A) 30Java

Java30

B) 30Java

Java1020

C) 1020Java

Java1020

D) 1020Java

Java30

Answer: B) 30Java

Java1020

Explanation: In the first `println`, `10 + 20` is evaluated first as integer addition, resulting in `30`. Then, `30` is concatenated with the string "Java". In the second `println`, evaluation proceeds from left to right. "Java" + 10 results in the string "Java10", which is then concatenated with 20 to produce "Java1020".

54. If a table has a column `OrderDate` of type `DATE`, which SQL clause would you use to select orders placed in the last 7 days?

A) WHERE OrderDate >= NOW() - INTERVAL '7' DAY

B) WHERE OrderDate > 7

C) HAVING OrderDate > GETDATE() - 7

D) WHERE OrderDate IN (LAST 7 DAYS)

Answer: A) WHERE OrderDate >= NOW() - INTERVAL '7' DAY

Explanation: The syntax for date functions varies between SQL dialects (e.g., `GETDATE()` in SQL Server, `NOW()` in MySQL/PostgreSQL). However, the general approach is to compare the `OrderDate` column to the current date minus a 7-day interval. Option A represents this common pattern.

55. In the context of algorithms, what does "Big O" notation represent?

A) The exact number of operations an algorithm will perform.

B) The average-case performance of an algorithm.

C) The worst-case time complexity or upper bound of an algorithm's growth rate.

D) The best-case performance of an algorithm.

Answer: C) The worst-case time complexity or upper bound of an algorithm's growth rate.

Explanation: Big O notation is used to classify algorithms according to how their run time or space requirements grow as the input size grows. It describes the upper bound on the growth rate, effectively representing the worst-case scenario.

56. What is the output of the C code below?

```
#include <stdio.h>
void func(char **ptr) {
   *ptr = "World";
}
int main() {
   char *str = "Hello";
   func(&str);
   printf("%s", str);
   return 0;
}
```

A) Hello

B) World

C) Compilation Error

D) Garbage Value

Answer: B) World

Explanation: The function `func` takes a pointer to a character pointer (`char **`). Inside `main`, the address of the pointer `str` is passed to `func`. The function then modifies the pointer `str` itself to point to the string literal "World". The `printf` in `main` then prints the new string `str` points to.

57. In C++, if you have a class without any constructors defined, what happens when you create an object?

A) The code fails to compile.

B) The object is created without any member initialization.

C) A default constructor is automatically provided by the compiler.

D) A runtime error occurs.

Answer: C) A default constructor is automatically provided by the compiler.

Explanation: If a class has no user-declared constructors, the compiler will declare a default constructor as an `inline public` member of its class. This constructor will have no arguments and an empty body.

58. What is the primary use of an `Inner Class` in Java?

A) To create multiple instances of a class within another class.

B) To logically group classes and interfaces in one place so that it can be more readable and maintainable.

C) To improve the performance of a program.

D) To allow a class to inherit from multiple classes.

Answer: B) To logically group classes and interfaces in one place so that it can be more readable and maintainable.

Explanation: Inner classes are a way of logically grouping classes that are only used in one place. It increases encapsulation, and an inner class has access to the members of its enclosing outer class, which can lead to more readable and maintainable code.

59. Which type of SQL join would you use to return all customers, and the orders they might have placed? (Including customers who have not placed any orders).

A) INNER JOIN

B) RIGHT JOIN

C) FULL OUTER JOIN

D) LEFT JOIN

Answer: D) LEFT JOIN

Explanation: Assuming `Customers` is the left table and `Orders` is the right table, a `LEFT JOIN` will return all rows from the `Customers` table, regardless of whether there is a match in the `Orders` table. For customers with no orders, the columns from the `Orders` table will be `NULL`.

60. What is a hash collision?

A) When two different hash functions produce the same output for the same key.

B) When a hash function produces the same output for two different keys.

C) When a hash table runs out of memory.

D) When a key cannot be hashed.

Answer: B) When a hash function produces the same output for two different keys.

Explanation: A hash collision occurs when the hash function maps two distinct input keys to the same output hash value. Since hash tables use this value as an index, collision resolution techniques (like chaining or open addressing) are needed to handle such cases.