Technical

1. What will be the output of the following C code?

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

A) 10

B) 11

C) 12

D) Undefined Behavior

Answer: D) Undefined Behavior

Explanation: The expression a++ + ++a causes undefined behavior. According to the C standard, modifying a variable more than once between two sequence points is not allowed. Here, a is modified by both the post-increment (a++) and pre-increment (++a) operators within the same expression without a sequence point, leading to an unpredictable result.

---

2. Predict the output of this C code snippet:

```
\#include \<stdio.h\>
int main() {
int arr[] = {10, 20, 30, 40, 50};
int \*ptr = arr;
printf("%d", \*(ptr++ + 2));
return 0;
}
```

A) 30

B) 20

C) Compiler Error

D) Garbage Value

Answer: A) 30

Explanation: The expression *(ptr++ + 2) is evaluated based on operator precedence. The post-increment ++ has higher precedence than +. However, the value of ptr used in the expression is its original value (pointing to arr[0]) before the increment happens. So, the expression becomes *(arr + 2), which points to the third element, 30. After the expression is evaluated, ptr is incremented to point to arr[1].

---

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

```
\#include \<stdio.h\>
int main() {
printf("%d", sizeof(void));
return 0;
}
```

A) 0

B) 1

C) 2

D) Compiler Error

Answer: D) Compiler Error

Explanation: The sizeof operator cannot be applied to an incomplete type like void. A void type has no size by definition, so the compiler will throw an error. However, sizeof(void*) is valid and would return the size of a pointer on the system.

---

4. What will be the output?

\#include \<stdio.h\>

\#define PRODUCT(x) (x*x)

int main() {

int i = 3;

int j = PRODUCT(i + 1);

printf("%d", j);

return 0;

}

A) 16

B) 10

C) 7

D) 12

Answer: C) 7

Explanation: Macros are simple text substitutions performed by the preprocessor. The line int j = PRODUCT(i + 1); will be expanded to int j = (i + 1 * i + 1);. Due to operator precedence (multiplication before addition), this is calculated as 3 + (1 * 3) + 1, which results in 3 + 3 + 1 = 7. To fix this, the macro should be defined as \#define PRODUCT(x) ((x)*(x)).

---

5. How many times will "Hello" be printed?

\#include \<stdio.h\>

int main() {

int i;

for (i = 0; i \< 5; i++);

printf("Hello");

return 0;

}

A) 5 times

B) 1 time

C) 0 times

D) Compiler Error

Answer: B) 1 time

Explanation: The semicolon (;) after the for loop statement creates an empty loop body. The loop runs 5 times, doing nothing in each iteration. After the loop condition i \< 5 becomes false (when i is 5), the loop terminates. The printf("Hello"); statement that follows is a separate statement and is executed only once.

---

6. What is the output of this code?

\#include \<stdio.h\>

int main() {

char str[] = "C-Program";

printf("%s %s\\n", str, str+2);

return 0;

}

A) C-Program C-Program

B) C-Program -Program

C) C- -Program

D) Compiler Error

Answer: B) C-Program -Program

Explanation: str is a pointer to the first character of the string. str+2 is a pointer to the third character of the string ('-'). When %s is used with printf, it prints characters starting from the given address until it encounters a null terminator (\\0). So, str prints the whole string, and str+2 prints the string from the third character onwards.

---

7. In C, if you pass an array as an argument to a function, what is actually passed?

A) The first element of the array

B) The base address of the array

C) A copy of the array

D) The size of the array

Answer: B) The base address of the array

Explanation: When an array is passed to a function, it "decays" into a pointer to its first element. This means the function receives the memory address of the beginning of the array, not a copy of the entire array. This is why modifications to the array inside the function affect the original array.

---

8. What is the purpose of the static keyword for a local variable in a function?

A) It makes the variable visible outside the function.

B) It allocates the variable on the heap instead of the stack.

C) It preserves the variable's value between function calls.

D) It prevents the variable from being modified.

Answer: C) It preserves the variable's value between function calls.

Explanation: A static local variable is initialized only once (the first time the function is called) and its lifetime extends across the entire run of the program. Its value is retained between successive calls to the function.

---

9. What is a dangling pointer?

A) A pointer that has not been initialized.

B) A pointer that points to a NULL value.

C) A pointer pointing to a memory location that has been freed.

D) A pointer that points to the starting address of an array.

Answer: C) A pointer pointing to a memory location that has been freed.

Explanation: A dangling pointer arises when a pointer continues to hold the address of a memory location that has been deallocated (e.g., via free() or because a local variable went out of scope). Accessing the memory through a dangling pointer leads to undefined behavior.

---

10. What will be the output of the following code?

```
\#include \<stdio.h\>
int main() {
int x = 10;
float y = 10.0;
if (x == y)
printf("Equal");
else
printf("Not Equal");
return 0;
}
```

A) Equal

B) Not Equal

C) Compiler Error

D) No output

Answer: A) Equal

Explanation: During the comparison x == y, the int value x is implicitly promoted to a float to match the type of y. The integer 10 is converted to the float 10.0, and the comparison 10.0 == 10.0 evaluates to true.

---

11. What is the size of a union?

A) The sum of the sizes of all its members.

B) The size of its largest member.

C) The size of its smallest member.

D) Always 4 bytes.

Answer: B) The size of its largest member.

Explanation: A union allocates enough memory to store only its largest member. All members of a union share the same memory location.

---

12. What does the volatile keyword signify?

A) The variable cannot be modified.

B) The variable's value can be changed by something beyond the compiler's control.

C) The variable is stored in a register for faster access.

D) The variable is a constant.

Answer: B) The variable's value can be changed by something beyond the compiler's control.

Explanation: The volatile keyword tells the compiler not to optimize the variable because its value may be changed by external factors (e.g., another thread, a hardware register). This ensures that every time the variable is accessed, its value is read directly from memory.

---

13. Which of the following is the correct way to open a file for both reading and writing in binary mode?

A) fopen("file.bin", "rw");

B) fopen("file.bin", "wb+");

C) fopen("file.bin", "r+w");

D) fopen("file.bin", "ab");

Answer: B) fopen("file.bin", "wb+");

Explanation: The modes for fopen are specific. "w" is for write, "r" is for read, "a" is for append. The "+" sign indicates updating (both reading and writing). The "b" indicates binary mode. wb+ creates a binary file for read/write. rb+ opens an existing binary file for read/write.

---

14. What will be the output of the following code?

\#include \<stdio.h\>

int main() {

int a = 1, b = 1, c;

c = a++ + b;

printf("a = %d, b = %d, c = %d", a, b, c);

return 0;

}

A) a = 2, b = 1, c = 2

B) a = 1, b = 1, c = 2

C) a = 2, b = 1, c = 3

D) a = 1, b = 2, c = 2

Answer: A) a = 2, b = 1, c = 2

Explanation: The expression a++ + b is evaluated as follows: the current value of a (which is 1) is used in the addition with b. So, c becomes 1 + 1 = 2. After the expression is evaluated, a is incremented to 2 due to the post-increment operator (a++).

---

15. Predict the output:

\#include \<stdio.h\>

int main() {

int x = 5;

printf("%d %d %d\\n", x, x \<\< 1, x \>\> 1);

return 0;

}

A) 5 10 2

B) 2 10 5

C) The output is compiler-dependent.

D) 5 2 10

Answer: C) The output is compiler-dependent.

Explanation: The order in which arguments are evaluated in a function call (like printf) is not specified by the C standard. A compiler could evaluate x, x \<\< 1, and x \>\> 1 in any order before passing them to the function. Therefore, the output can vary.

---

16. What is the role of extern keyword?

A) It declares a variable without defining it.

B) It gives a variable static storage duration.

C) It makes a variable local to a file.

D) It initializes a variable to zero.

Answer: A) It declares a variable without defining it.

Explanation: The extern keyword is used to provide a declaration of a global variable that is defined in another file. It tells the compiler that the variable exists but is defined elsewhere, allowing different source files to share the same global variable.

---

17. What does typedef do?

A) Creates a new data type.

B) Creates an alias for an existing data type.

C) Defines a new variable.

D) Declares a function.

Answer: B) Creates an alias for an existing data type.

Explanation: typedef is used to create a new name (alias) for an existing data type. This is often used to make complex type declarations, like for structs or function pointers, more readable.

---

18. What will be the value of x?

```
\#include \<stdio.h\>

int main() {

int x = 10;

x = x++;

printf("%d", x);

return 0;

}
```

A) 10

B) 11

C) 9

D) Undefined Behavior

Answer: D) Undefined Behavior

Explanation: Like in question 1, this statement x = x++ modifies the variable x twice between sequence points (once by the assignment, once by the increment). This results in undefined behavior.

---

19. What is the difference between malloc() and calloc()?

A) malloc() allocates memory for a single item, calloc() for multiple.

B) malloc() initializes allocated memory to zero, calloc() does not.

C) calloc() initializes allocated memory to zero, malloc() does not.

D) There is no difference.

Answer: C) calloc() initializes allocated memory to zero, malloc() does not.

Explanation: Both malloc() and calloc() allocate memory from the heap. The key difference is that malloc() leaves the allocated memory uninitialized (containing garbage values), while calloc() initializes all bytes in the allocated memory block to zero.

---

20. What is the output?

```
\#include \<stdio.h\>
int main() {
int x = 3;
switch (x) {
case 1: printf("One ");
case 2: printf("Two ");
case 3: printf("Three ");
case 4: printf("Four ");
default: printf("Default");
}
return 0;
}
```

A) Three

B) Three Default

C) Three Four Default

D) Compiler Error

Answer: C) Three Four Default

Explanation: The switch statement in C exhibits "fall-through" behavior. Once a matching case is found (case 3), execution starts from there and continues through all subsequent case statements (and the default block) until a break statement or the end of the switch block is encountered. Since there are no break statements, it prints "Three ", "Four ", and "Default".

---

21. A pointer that is pointing to NOTHING is called a _____ pointer.

A) VOID

B) DANGLING

C) NULL

D) WILD

Answer: C) NULL

Explanation: A NULL pointer is a special pointer that is guaranteed not to point to any valid memory location. It's often used to indicate the absence of a value or the end of a data structure (like a linked list).

---

22. How are command-line arguments passed to main()?

A) int main(int argv, char *argc)

B) int main(char *argv[], int argc)

C) int main(int argc, char *argv[])

D) int main(char argc, int *argv)

Answer: C) int main(int argc, char *argv[])

Explanation: The standard signature for main to accept command-line arguments is int main(int argc, char *argv[]). argc (argument count) is an integer representing the number of arguments, and argv (argument vector) is an array of character pointers (strings), where each string is one of the arguments.

---

23. What is endianness?

A) The order in which bits are arranged in a byte.

B) The order in which bytes are stored in multi-byte data types.

C) The way floating-point numbers are represented.

D) The maximum value an integer can hold.

Answer: B) The order in which bytes are stored in multi-byte data types.

Explanation: Endianness refers to the byte order. Big-endian stores the most significant byte at the smallest memory address. Little-endian stores the least significant byte at the smallest memory address.

---

24. Which header file is required for using strcmp()?

A) \<stdio.h\>

B) \<stdlib.h\>

C) \<string.h\>

D) \<conio.h\>

Answer: C) \<string.h\>

Explanation: The \<string.h\> header file contains declarations for string manipulation functions in C, including strcpy, strcat, strlen, and strcmp.

---

25. Predict the output:

```
\#include \<stdio.h\>
int main() {
float f = 0.1;
if (f == 0.1)
printf("Equal");
else
printf("Not Equal");
return 0;
}
```

A) Equal

B) Not Equal

C) Sometimes Equal, sometimes Not Equal

D) Compiler Error

Answer: B) Not Equal

Explanation: This is a classic floating-point precision issue. The literal 0.1 is a double by default. The float variable f stores an approximation of 0.1. Due to the different internal binary representations

of float and double, the comparison f == 0.1 is likely to be false. It's better to check if the absolute difference is within a small tolerance: if (fabs(f - 0.1) \< 0.00001).

---

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

```
\#include \<iostream\>
class Base {
public:
virtual void show() { std::cout \<\< "Base "; }
};
class Derived : public Base {
public:
void show() { std::cout \<\< "Derived "; }
};
int main() {
Base \*b = new Derived();
b-\>show();
delete b;
return 0;
}
```

A) Base

B) Derived

C) Compiler Error

D) Runtime Error

Answer: B) Derived

Explanation: Since show() is a virtual function in the Base class, the call b-\>show() is resolved at runtime (late binding). The pointer b is of type Base*, but it points to an object of type Derived. Therefore, the Derived class's version of show() is executed. This is a core concept of runtime polymorphism.

---

27. What is the output of this code snippet?

```
\#include \<iostream\>
class Parent {
public:
Parent() { std::cout \<\< "Parent Constructor "; }
\~Parent() { std::cout \<\< "Parent Destructor "; }
};
class Child : public Parent {
public:
Child() { std::cout \<\< "Child Constructor "; }
\~Child() { std::cout \<\< "Child Destructor "; }
};
int main() {
Child c;
return 0;
}
```

A) Parent Constructor Child Constructor Parent Destructor Child Destructor

B) Child Constructor Parent Constructor Child Destructor Parent Destructor

C) Parent Constructor Child Constructor Child Destructor Parent Destructor

D) Child Constructor Parent Constructor Parent Destructor Child Destructor

Answer: C) Parent Constructor Child Constructor Child Destructor Parent Destructor

Explanation: When an object of a derived class is created, the base class constructor is always called first, followed by the derived class constructor. When the object is destroyed, the order is reversed: the derived class destructor is called first, followed by the base class destructor.

---

28. Which concept is demonstrated by having two functions with the same name but different parameter lists in the same class?

A) Function Overriding

B) Function Overloading

C) Operator Overloading

D) Polymorphism

Answer: B) Function Overloading

Explanation: Function overloading allows multiple functions in the same scope to have the same name, as long as their parameter lists differ in number or type. The compiler resolves which function to call based on the arguments provided at compile time.

---

29. What is the potential issue in this code?

\#include \<iostream\>

class MyClass {

public:

MyClass() { std::cout \<\< "Constructed\\n"; }

\~MyClass() { std::cout \<\< "Destructed\\n"; }

};

int main() {

MyClass *p = new MyClass[2];

delete p;

return 0;

}

A) No issue, it works fine.

B) It will cause a compile-time error.

C) It will result in a memory leak and undefined behavior.

D) It will print "Destructed" twice.

Answer: C) It will result in a memory leak and undefined behavior.

Explanation: When allocating an array of objects with new[], you must use delete[] to deallocate it. Using delete p; instead of delete[] p; results in undefined behavior. Typically, only the destructor for

the first object in the array is called, and the memory for the subsequent objects is not properly released, causing a memory leak.

---

30. What does std::vector\<int\>::size() return?

A) The number of elements currently in the vector.

B) The total number of elements the vector can hold before reallocation.

C) The size of the vector in bytes.

D) A pointer to the last element.

Answer: A) The number of elements currently in the vector.

Explanation: size() returns the actual number of elements stored in the vector. capacity() returns the storage space currently allocated for the vector, which can be greater than or equal to size().

---

31. What is the primary difference between a struct and a class in C++?

A) struct members are public by default, while class members are private by default.

B) struct cannot have member functions, while class can.

C) struct is a value type, while class is a reference type.

D) There is no difference.

Answer: A) struct members are public by default, while class members are private by default.

Explanation: This is the only difference between a struct and a class in C++. In all other respects (inheritance, member functions, constructors, etc.), they are functionally identical.

---

32. What will be the output of this code?

```
\#include \<iostream\>
void modify(int& val) {
```

```
val = 100;
}
int main() {
int x = 50;
modify(x);
std::cout \<\< x;
return 0;
}
```

A) 50

B) 100

C) Garbage Value

D) Compiler Error

Answer: B) 100

Explanation: The function modify takes its argument by reference (int& val). This means val becomes an alias for the variable x passed from main. Any modification to val inside the function directly changes the original variable x.

---

33. What is the role of the this pointer?

A) It points to the base class of the current object.

B) It points to the current object itself.

C) It is a void pointer that can point to any object.

D) It points to a static member of the class.

Answer: B) It points to the current object itself.

Explanation: Inside a non-static member function, the this keyword is a pointer to the object on which the member function was called. It is used to access member variables when they are shadowed by local variables or to return a reference to the current object.

---

34. In the context of exception handling, what does the throw keyword do?

A) It catches an exception.

B) It specifies a block of code to be tested for errors.

C) It signals that an exceptional condition has occurred.

D) It defines a block of code to be executed if an error occurs.

Answer: C) It signals that an exceptional condition has occurred.

Explanation: The throw statement is used to signal an exception. It is followed by an argument (the exception object) which is then passed to an exception handler (catch block).

---

35. Which smart pointer should be used for exclusive ownership of a resource?

A) std::shared_ptr

B) std::weak_ptr

C) std::unique_ptr

D) std::auto_ptr (deprecated)

Answer: C) std::unique_ptr

Explanation: std::unique_ptr is a smart pointer that owns and manages another object through a pointer and disposes of that object when the unique_ptr goes out of scope. It enforces exclusive ownership, meaning it cannot be copied, only moved.

---

36. What is RAII (Resource Acquisition Is Initialization)?

A) A design pattern for managing resources in C++.

B) A memory allocation technique.

C) A type of inheritance.

D) A template metaprogramming technique.

Answer: A) A design pattern for managing resources in C++.

Explanation: RAII is a programming idiom where resource management (like memory, file handles, sockets) is tied to the lifetime of objects. The resource is acquired in the object's constructor and released in its destructor. This helps prevent resource leaks, especially in the presence of exceptions. Smart pointers like std::unique_ptr are prime examples of RAII.

---

37. What is the output of this code?

```
\#include \<iostream\>
struct A {
A() { std::cout \<\< "A"; }
A(const A& other) { std::cout \<\< "Ac"; }
};
void func(A val) {}
int main() {
A a;
func(a);
return 0;
}
```

A) A

B) AAc

C) Ac

D) AA

Answer: B) AAc

Explanation: First, the object a is created, so the default constructor is called, printing "A". Then, a is passed by value to the function func. This creates a copy of a, so the copy constructor is called, printing "Ac".

---

38. What is the 'Rule of Three' in C++?

A) If a class defines a constructor, it should also define an empty constructor and a destructor.

B) A class should have at most three base classes.

C) If a class defines a destructor, a copy constructor, or a copy assignment operator, it should probably define all three.

D) A function should not have more than three arguments.

Answer: C) If a class defines a destructor, a copy constructor, or a copy assignment operator, it should probably define all three.

Explanation: The 'Rule of Three' (now often extended to the 'Rule of Five' in modern C++) is a rule of thumb for classes that manage resources. If a class needs a user-defined destructor (e.g., to free memory), it almost certainly needs a user-defined copy constructor and a user-defined copy assignment operator to handle resource copying correctly and avoid issues like shallow copies and double frees.

---

39. What does the const keyword at the end of a member function declaration signify? void print() const;

A) The function returns a constant value.

B) The function cannot be overridden.

C) The function cannot modify any member variables of the object.

D) The function can only be called on constant objects.

Answer: C) The function cannot modify any member variables of the object.

Explanation: A const member function promises not to change the state of the object it is called on. Inside such a function, the this pointer is a pointer to const, and any attempt to modify a member variable will result in a compile-time error.

---

40. What is a V-Table (Virtual Table)?

A) A table used to store variables.

B) A mechanism used to achieve dynamic polymorphism.

C) A table of contents for a class.

D) A static table shared by all objects of a class.

Answer: B) A mechanism used to achieve dynamic polymorphism.

Explanation: A V-Table is a lookup table of function pointers used to resolve virtual function calls at runtime. Each class that has virtual functions has its own V-Table, and each object of such a class has a hidden pointer (vptr) to that table.

---

41. What is the output?

```
\#include \<iostream\>
\#include \<vector\>
int main() {
std::vector\<int\> v = {1, 2, 3};
std::vector\<int\>& ref = v;
ref.push\_back(4);
std::cout \<\< v.size();
return 0;
}
```

A) 3

B) 4

C) 0

D) Compiler Error

Answer: B) 4

Explanation: ref is a reference to the vector v. It is not a copy but an alias. Modifying ref by calling push_back(4) is the same as modifying v. Therefore, the size of v becomes 4.

---

42. Which of the following is true about static member variables?

A) Each object of the class has its own copy.

B) They can only be accessed by static member functions.

C) They are shared among all objects of the class.

D) They are initialized by the constructor.

Answer: C) They are shared among all objects of the class.

Explanation: A static member variable belongs to the class itself, not to any individual object. There is only one copy of the static variable, which is shared by all instances of the class. It must be defined and initialized outside the class.

---

43. What is function overriding?

A) Two functions with the same name and different parameters.

B) A derived class providing a specific implementation of a base class function.

C) Creating a template function.

D) Deleting a function from the base class.

Answer: B) A derived class providing a specific implementation of a base class function.

Explanation: Function overriding occurs when a derived class creates a function with the same name, parameters, and return type as a function in its base class. If the base class function is virtual, this allows for polymorphic behavior.

---

44. What is the purpose of the friend keyword?

A) To allow a function or another class to access the private and protected members of the class where it is declared.

B) To specify inheritance.

C) To create an alias for a class.

D) To make a member variable constant.

Answer: A) To allow a function or another class to access the private and protected members of the class where it is declared.

Explanation: A friend declaration breaks encapsulation by granting a non-member function or another class full access to the private and protected members of the declaring class.

---

45. What is std::move used for?

A) To move a file on the filesystem.

B) To cast a value to an rvalue reference, enabling move semantics.

C) To reallocate memory for an object.

D) To copy an object to a new memory location.

Answer: B) To cast a value to an rvalue reference, enabling move semantics.

Explanation: std::move doesn't actually move anything. It's a cast that produces an xvalue (a type of rvalue). This signals that the object's resources can be "stolen" or "pilfered" by another object, which is the basis of move semantics, a C++11 optimization that avoids expensive copies.

---

46. What is the difference between delete and delete[]?

A) delete is for pointers, delete[] is for arrays.

B) delete calls one destructor, delete[] calls the destructor for every element in the array.

C) delete[] is faster than delete.

D) Both A and B.

Answer: D) Both A and B.

Explanation: You must use the operator that corresponds to the allocation. new is paired with delete. new[] is paired with delete[]. Using delete on a pointer allocated with new[] results in undefined behavior because it typically only calls the destructor for the first element, leading to resource leaks.

---

47. Which of these is a sequence container in STL?

A) std::set

B) std::map

C) std::vector

D) std::unordered_map

Answer: C) std::vector

Explanation: Sequence containers in the C++ Standard Template Library (STL) store elements in a linear sequence. Examples include std::vector, std::deque, std::list, and std::array. std::set and std::map are associative containers.

---

48. What is a lambda expression in C++11?

A) A way to define an anonymous function object.

B) A type of preprocessor macro.

C) A keyword for declaring variables.

D) A template specialization.

Answer: A) A way to define an anonymous function object.

Explanation: A lambda expression provides a convenient way to define an unnamed function object (a functor) right at the location where it is needed. It's especially useful for algorithms that take functions as arguments, like std::sort or std::for_each.

---

49. What is the scope resolution operator?

A) .

B) -\>

C) &

D) ::

Answer: D) ::

Explanation: The scope resolution operator (::) is used to access static members, to define member functions outside a class, to access members of a namespace, or to refer to global variables that have been hidden by local variables of the same name.

---

50. What is a pure virtual function?

A) A virtual function that is initialized to 0.

B) A function that cannot be overridden.

C) A virtual function that has no implementation in the base class.

D) A function that can only be called from a derived class.

Answer: C) A virtual function that has no implementation in the base class.

Explanation: A pure virtual function is declared by assigning = 0 in its declaration, like virtual void draw() = 0;. This makes the class an abstract class, which cannot be instantiated. Any derived class must implement all pure virtual functions of its base class to become a concrete (instantiable) class.

---

51. What is the output of the following Java code?

```
public class Test {
public static void main(String[] args) {
String s1 = "Java";
String s2 = new String("Java");
String s3 = "Java";

```

```
System.out.print(s1 == s2);
System.out.print(" ");
System.out.print(s1 == s3);
```

```
}
}
```

A) true true

B) false true

C) true false

D) false false

Answer: B) false true

Explanation: s1 refers to an object in the String Constant Pool. s2 is created using the new keyword, so it refers to a new object on the heap. s1 == s2 compares memory addresses, which are different, so it's false. s3 is assigned the same literal as s1, so the JVM makes it point to the same object in the String Pool. Therefore, s1 == s3 is true.

---

52. What will happen when this code is compiled and run?

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

A) ACDE

B) ABDE

C) AC

D) ADE

Answer: A) ACDE

Explanation: The code first prints "A". The line 5 / 0 throws an ArithmeticException. The catch block for this exception is executed, printing "C". The finally block is always executed, regardless of whether an exception occurred, so it prints "D". Finally, the code after the try-catch-finally block executes, printing "E".

---

53. Which of the following is NOT a core concept of Object-Oriented Programming?

A) Encapsulation

B) Inheritance

C) Polymorphism

D) Procedure

Answer: D) Procedure

Explanation: Encapsulation, Inheritance, and Polymorphism are three of the four main pillars of OOP (the fourth being Abstraction). A procedure is a concept from procedural programming, which is a different programming paradigm.

---

54. What is the difference between an abstract class and an interface in Java (pre-Java 8)?

A) An abstract class can have instance variables, while an interface cannot.

B) An interface can have constructors.

C) A class can extend multiple abstract classes.

D) An abstract class can only have abstract methods.

Answer: A) An abstract class can have instance variables, while an interface cannot.

Explanation: Prior to Java 8, interfaces could only have public static final variables and abstract methods. Abstract classes can have instance variables (state), concrete methods, and constructors, in addition to abstract methods.

---

55. What is the output of this program?

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

A) [A, B, C]

B) [A, C, B]

C) [C, A, B]

D) [A, B]

Answer: B) [A, C, B]

Explanation: list.add("A") adds "A" at index 0. list.add("B") adds "B" at index 1. The list is now [A, B]. The method list.add(1, "C") inserts the element "C" at the specified index 1. It shifts the element currently at that position ("B") and any subsequent elements to the right. The final list is [A, C, B].

---

56. Which statement about the final keyword in Java is FALSE?

A) A final class cannot be subclassed.

B) A final method cannot be overridden.

C) The value of a final variable cannot be changed.

D) A final method cannot be overloaded.

Answer: D) A final method cannot be overloaded.

Explanation: The final keyword has no effect on method overloading. A method can be overloaded regardless of whether it is declared final. Overloading is about different method signatures (name + parameters), while final relates to preventing overriding in subclasses.

---

57. What is the initial value of an uninitialized instance variable of type int and String?

A) 0 and ""

B) 0 and null

C) null and null

D) Compilation Error

Answer: B) 0 and null

Explanation: In Java, instance variables are given default values if not explicitly initialized. For numeric primitives like int, the default is 0. For object references like String, the default is null.

---

58. What is the primary purpose of garbage collection in Java?

A) To remove unused classes from the classpath.

B) To clean up the disk by deleting temporary files.

C) To automatically free up memory occupied by objects that are no longer in use.

D) To close unclosed database connections.

Answer: C) To automatically free up memory occupied by objects that are no longer in use.

Explanation: The garbage collector (GC) is a form of automatic memory management. It runs in the background and deallocates memory for objects that are no longer reachable from any part of the running program, thus preventing memory leaks.

---

59. Which collection class stores elements in a key-value pair format and does not allow duplicate keys?

A) ArrayList

B) HashSet

C) HashMap

D) TreeMap

Answer: C) HashMap

Explanation: HashMap is an implementation of the Map interface. It stores data as key-value pairs. Each key must be unique; if you try to insert a duplicate key, the old value associated with that key will be replaced.

---

60. How do you create a thread in Java?

A) By extending the Thread class or implementing the Runnable interface.

B) By creating an instance of the Process class.

C) By calling the start() method on any Object.

D) By implementing the Serializable interface.

Answer: A) By extending the Thread class or implementing the Runnable interface.

Explanation: These are the two standard ways to define the code that will be executed in a new thread. Implementing Runnable is often preferred as it allows the class to extend another class, avoiding the limitations of single inheritance in Java.

---

61. Which of these is a checked exception?

A) RuntimeException

B) ArithmeticException

C) NullPointerException

D) IOException

Answer: D) IOException

Explanation: In Java, exceptions are categorized as checked or unchecked. Checked exceptions are exceptions that a method must either handle (with a try-catch block) or declare in its throws clause. IOException and its subclasses are checked exceptions. RuntimeException and its subclasses (like ArithmeticException and NullPointerException) are unchecked exceptions, and the compiler does not require you to handle them.

---

62. What is the result of System.out.println("a" + 'b' + 3);?

A) ab3

B) 100

C) a98

D) Compilation Error

Answer: A) ab3

Explanation: The + operator, when used with a String on either side, performs string concatenation. The expression is evaluated from left to right. "a" + 'b' results in the string "ab". Then, "ab" + 3 results in the string "ab3".

---

63. Can a Java source file have more than one public class?

A) Yes, if they are in the same package.

B) No.

C) Yes, if they are inner classes.

D) Yes, if the filename is AllClasses.java.

Answer: B) No.

Explanation: A Java source file can have at most one public top-level class. If a public class exists, the name of the source file must match the name of the public class. A source file can contain multiple non-public classes.

---

64. What is the contract between equals() and hashCode()?

A) If two objects are equal according to equals(), they must have the same hashCode().

B) If two objects have the same hashCode(), they must be equal according to equals().

C) There is no contract between them.

D) Both A and B are true.

Answer: A) If two objects are equal according to equals(), they must have the same hashCode().

Explanation: This is the fundamental contract. The reverse is not true: two objects with the same hashCode() are not required to be equal (this is a hash collision). This contract is crucial for the correct functioning of hash-based collections like HashMap and HashSet.

---

65. Which of the following access modifiers provides the widest accessibility?

A) private

B) default (package-private)

C) protected

D) public

Answer: D) public

Explanation: public members are accessible from anywhere. protected members are accessible within the package and by subclasses. default members are accessible only within the same package. private members are accessible only within the same class.

---

66. What is method overriding?

A) A class having two methods with the same name but different parameters.

B) A subclass providing a specific implementation for a method that is already defined in its superclass.

C) Calling a method from a superclass.

D) A method that cannot be inherited.

Answer: B) A subclass providing a specific implementation for a method that is already defined in its superclass.

Explanation: Method overriding is a key feature of polymorphism. The method in the subclass must have the same name, return type, and parameters as the one in the superclass.

---

67. What is autoboxing?

A) The automatic conversion of a primitive type to its corresponding wrapper class object.

B) The automatic conversion of a wrapper class object to its corresponding primitive type.

C) A technique for boxing up objects for serialization.

D) Encapsulating a variable within a class.

Answer: A) The automatic conversion of a primitive type to its corresponding wrapper class object.

Explanation: For example, converting an int to an Integer automatically, like Integer i = 100;. The reverse process is called unboxing.

---

68. What will be the output?

```
import java.util.HashSet;
import java.util.Set;
public class Test {
public static void main(String[] args) {
Set\<String\> set = new HashSet\<\>();
set.add("A");
```

```
set.add("B");
boolean isAdded = set.add("A");
System.out.println(set.size() + " " + isAdded);
}
}
```

A) 3 true

B) 2 true

C) 2 false

D) 3 false

Answer: C) 2 false

Explanation: A HashSet only stores unique elements. The first set.add("A") is successful. The second time set.add("A") is called, the element "A" is already present, so it is not added again. The add method returns false if the element is already in the set. The final size of the set is 2.

---

69. What is the super keyword used for?

A) To refer to the current instance of the class.

B) To invoke a superclass's constructor or members.

C) To define a class that cannot be extended.

D) To declare a static variable.

Answer: B) To invoke a superclass's constructor or members.

Explanation: super is used to differentiate members (fields or methods) of the superclass from the members of the subclass if they have the same name. super() is used to call the constructor of the parent class.

---

70. Which of these is a functional interface?

A) An interface with exactly one method.

B) An interface with exactly one abstract method.

C) An interface with only static methods.

D) An interface that extends Runnable.

Answer: B) An interface with exactly one abstract method.

Explanation: A functional interface is an interface that contains only one abstract method. They are used as the basis for lambda expressions and method references in Java 8 and later. The @FunctionalInterface annotation can be used to enforce this.

---

71. What does it mean for a method to be synchronized?

A) It can only be called once.

B) It ensures that only one thread can execute that method on a given object instance at a time.

C) It synchronizes the object's data with a database.

D) It can only be accessed by other synchronized methods.

Answer: B) It ensures that only one thread can execute that method on a given object instance at a time.

Explanation: The synchronized keyword provides a lock on the object. When a thread enters a synchronized method, it acquires the lock, and no other thread can enter any synchronized method on the same object instance until the first thread releases the lock.

---

72. What is the main difference between String, StringBuilder, and StringBuffer?

A) String is mutable; StringBuilder and StringBuffer are immutable.

B) String is immutable; StringBuilder and StringBuffer are mutable.

C) StringBuffer is not thread-safe.

D) StringBuilder is thread-safe.

Answer: B) String is immutable; StringBuilder and StringBuffer are mutable.

Explanation: String objects cannot be changed after creation. StringBuilder and StringBuffer objects can be modified. The difference between the latter two is that StringBuffer methods are synchronized (thread-safe), while StringBuilder methods are not, making StringBuilder faster in single-threaded environments.

---

73. What is the purpose of the transient keyword?

A) To make a variable's value temporary.

B) To prevent a field from being serialized when its object is serialized.

C) To mark a variable as thread-local.

D) To make a variable volatile.

Answer: B) To prevent a field from being serialized when its object is serialized.

Explanation: When an object is written to a stream (serialized), the transient keyword indicates that a particular instance variable should not be saved as part of the object's persistent state.

---

74. What is the output of the following code?

```
public class Test {
public static void main(String[] args) {
Integer i1 = 127;
Integer i2 = 127;
Integer i3 = 128;
Integer i4 = 128;
System.out.println(i1 == i2);
System.out.println(i3 == i4);
}
}
```

A) true true

B) false false

C) true false

D) false true

Answer: C) true false

Explanation: Java caches Integer objects for values from -128 to 127. When i1 and i2 are created via autoboxing, they both point to the same cached object. So i1 == i2 is true. The value 128 is outside this range, so new Integer objects are created for i3 and i4, and they have different memory addresses. Thus, i3 == i4 is false.

---

75. The finally block will not be executed if:

A) An exception is not thrown.

B) return is used in the try block.

C) The thread is killed or System.exit() is called.

D) An exception is thrown and caught.

Answer: C) The thread is killed or System.exit() is called.

Explanation: The finally block is designed to execute under almost all circumstances, including normal exit, exceptions, and return statements. The only common ways it won't execute are if the JVM is terminated abruptly via System.exit() or if the thread executing the code dies.

---

76. Which query finds the number of employees in each department? (Scenario: Employees and Departments tables)

A) SELECT DeptName, SUM(EmployeeID) FROM Employees JOIN Departments ON Employees.DeptID = Departments.DeptID;

B) SELECT DeptName, COUNT(*) FROM Employees;

C) SELECT d.DeptName, COUNT(e.EmployeeID) FROM Employees e JOIN Departments d ON e.DeptID = d.DeptID GROUP BY d.DeptName;

D) SELECT DeptName, COUNT(EmployeeID) FROM Employees GROUP BY DeptID;

Answer: C) SELECT d.DeptName, COUNT(e.EmployeeID) FROM Employees e JOIN Departments d ON e.DeptID = d.DeptID GROUP BY d.DeptName;

Explanation: To get the department name and the employee count, you must JOIN the two tables on their common key (DeptID). Then, you GROUP BY the department name and use an aggregate function like COUNT() to count the employees in each group.

---

77. Which query selects all employees whose last name starts with the letter 'S'?

A) SELECT * FROM Employees WHERE LastName = 'S%';

B) SELECT * FROM Employees WHERE LastName LIKE 'S%';

C) SELECT * FROM Employees WHERE LastName LIKE 'S_';

D) SELECT * FROM Employees WHERE LastName = 'S';

Answer: B) SELECT * FROM Employees WHERE LastName LIKE 'S%';

Explanation: The LIKE operator is used for pattern matching in strings. The wildcard character % matches any sequence of zero or more characters. Therefore, 'S%' matches any string that starts with 'S'.

---

78. What is the difference between DELETE and TRUNCATE?

A) DELETE is DML, TRUNCATE is DDL; TRUNCATE is faster and generally cannot be rolled back.

B) DELETE removes all rows, TRUNCATE removes only specified rows.

C) TRUNCATE activates triggers, DELETE does not.

D) There is no difference.

Answer: A) DELETE is DML, TRUNCATE is DDL; TRUNCATE is faster and generally cannot be rolled back.

Explanation: DELETE is a DML (Data Manipulation Language) command that removes rows one by one and logs each deletion. It can be rolled back and can have a WHERE clause. TRUNCATE is a DDL (Data Definition Language) command that deallocates the data pages of the table, making it much faster. It cannot have a WHERE clause and does not fire DELETE triggers.

---

79. Which query returns departments with an average employee salary (from a Salary column in Employees) greater than 50000?

A) SELECT DeptID FROM Employees WHERE Salary \> 50000 GROUP BY DeptID;

B) SELECT DeptID FROM Employees GROUP BY DeptID WHERE AVG(Salary) \> 50000;

C) SELECT DeptID FROM Employees GROUP BY DeptID HAVING AVG(Salary) \> 50000;

D) SELECT DeptID FROM Employees HAVING AVG(Salary) \> 50000;

Answer: C) SELECT DeptID FROM Employees GROUP BY DeptID HAVING AVG(Salary) \> 50000;

Explanation: The WHERE clause is used to filter rows before any groupings are made. To filter groups based on the result of an aggregate function (like AVG(), COUNT(), etc.), you must use the HAVING clause after the GROUP BY clause.

---

80. What does an INNER JOIN between Employees and Departments return?

A) All employees, regardless of whether they have a department.

B) All departments, regardless of whether they have employees.

C) Only the employees who are assigned to a department that exists in the Departments table.

D) All rows from both tables, combined.

Answer: C) Only the employees who are assigned to a department that exists in the Departments table.

Explanation: An INNER JOIN (or just JOIN) returns a result set containing only the rows that have matching values in both tables based on the specified join condition. Rows from either table that do not have a corresponding match in the other table are excluded.

---

81. Which type of constraint ensures that all values in a column are different?

A) PRIMARY KEY

B) UNIQUE

C) FOREIGN KEY

D) CHECK

Answer: B) UNIQUE

Explanation: The UNIQUE constraint ensures that no two rows have the same value in that column. A PRIMARY KEY also enforces uniqueness, but it has the additional constraint of not allowing NULL values. A table can have multiple UNIQUE constraints but only one PRIMARY KEY.

---

82. What is the purpose of an index in a database?

A) To enforce data integrity rules.

B) To provide a unique identifier for each row.

C) To speed up the retrieval of rows from a table.

D) To create a backup of the table data.

Answer: C) To speed up the retrieval of rows from a table.

Explanation: An index is a special lookup table that the database search engine can use to find data more quickly. It improves the speed of SELECT queries and WHERE clauses but can slow down data modification operations (INSERT, UPDATE, DELETE) because the index also needs to be updated.

---

83. What is the difference between UNION and UNION ALL?

A) UNION includes duplicate rows, while UNION ALL removes them.

B) UNION ALL includes duplicate rows, while UNION removes them.

C) UNION works on columns, while UNION ALL works on rows.

D) There is no difference; ALL is an optional keyword.

Answer: B) UNION ALL includes duplicate rows, while UNION removes them.

Explanation: Both operators combine the result sets of two or more SELECT statements. UNION performs a DISTINCT operation on the combined result set to remove duplicate rows. UNION ALL

simply concatenates the result sets, including all duplicates. UNION ALL is faster because it does not have to check for duplicates.

---

84. A Primary Key can be made of a single column or multiple columns. What is a key made of multiple columns called?

A) Foreign Key

B) Super Key

C) Composite Key

D) Candidate Key

Answer: C) Composite Key

Explanation: A composite key (or compound key) is a primary key that consists of two or more columns. The combination of these columns' values must be unique for each row in the table.

---

85. Which normal form deals with the elimination of transitive dependencies?

A) First Normal Form (1NF)

B) Second Normal Form (2NF)

C) Third Normal Form (3NF)

D) Boyce-Codd Normal Form (BCNF)

Answer: C) Third Normal Form (3NF)

Explanation: A table is in 3NF if it is in 2NF and has no transitive dependencies. A transitive dependency exists when a non-key attribute is functionally dependent on another non-key attribute. 3NF aims to reduce data redundancy by ensuring that attributes depend only on the primary key.

---

86. Which SQL statement is used to add a new column 'HireDate' of type DATE to the 'Employees' table?

A) ADD COLUMN HireDate DATE TO Employees;

B) MODIFY TABLE Employees ADD HireDate DATE;

C) ALTER TABLE Employees ADD COLUMN HireDate DATE;

D) UPDATE TABLE Employees ADD HireDate DATE;

Answer: C) ALTER TABLE Employees ADD COLUMN HireDate DATE;

Explanation: The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ADD COLUMN clause specifies the new column's name and data type. (Note: some SQL dialects allow ADD HireDate DATE without the COLUMN keyword).

---

87. What is a view in SQL?

A) A backup copy of a table.

B) A special type of table that can store images.

C) A virtual table based on the result-set of an SQL statement.

D) An index on a table.

Answer: C) A virtual table based on the result-set of an SQL statement.

Explanation: A view is a stored query that can be treated as a table. It contains rows and columns, just like a real table, but it does not store data itself. The data is generated dynamically when the view is queried. Views are often used to simplify complex queries, restrict access to data, or present data in a different format.

---

88. Which join returns all rows from the left table and the matched rows from the right table, with NULL for the columns from the right table where there is no match?

A) INNER JOIN

B) LEFT JOIN

C) RIGHT JOIN

D) FULL OUTER JOIN

Answer: B) LEFT JOIN

Explanation: A LEFT JOIN (or LEFT OUTER JOIN) returns all records from the left table and the matching records from the right table. If there is no match for a record in the left table, the result is NULL on the right side.

---

89. What are ACID properties in the context of transactions?

A) Atomicity, Consistency, Isolation, Durability

B) Authorization, Concurrency, Integrity, Data

C) Asynchronous, Committed, Isolated, Distributed

D) Atomic, Concurrent, Inconsistent, Durable

Answer: A) Atomicity, Consistency, Isolation, Durability

Explanation: ACID is a set of properties that guarantee database transactions are processed reliably. Atomicity: all operations in a transaction succeed or all of them fail. Consistency: the database remains in a valid state. Isolation: concurrent transactions do not interfere with each other. Durability: once a transaction is committed, it will remain so.

---

90. Which of the following is NOT an aggregate function in SQL?

A) COUNT()

B) MAX()

C) TRIM()

D) AVG()

Answer: C) TRIM()

Explanation: TRIM() is a scalar function (or string function) that removes leading and trailing spaces from a string. Aggregate functions like COUNT(), MAX(), and AVG() operate on a set of rows and return a single summary value.

---

91. What is a subquery?

A) A query that is nested inside another query.

B) A query that uses a UNION clause.

C) A query that is not valid.

D) A query that creates a temporary table.

Answer: A) A query that is nested inside another query.

Explanation: A subquery, or inner query, is a SELECT statement that is embedded inside another SQL statement (like SELECT, INSERT, UPDATE, or DELETE, or inside another subquery).

---

92. The ORDER BY clause sorts the result set in which order by default?

A) Descending (DESC)

B) Random

C) It depends on the primary key.

D) Ascending (ASC)

Answer: D) Ascending (ASC)

Explanation: Unless specified otherwise using the DESC keyword, the ORDER BY clause will sort the records in ascending order.

---

93. Which statement is used to give a user permission to work with a database object?

A) ALLOW

B) GRANT

C) PERMIT

D) REVOKE

Answer: B) GRANT

Explanation: GRANT is the SQL command used to provide privileges (like SELECT, INSERT, UPDATE, etc.) to a database user on specific database objects like tables or views. REVOKE is used to remove those privileges.

---

94. What does the COALESCE function do?

A) Concatenates two strings.

B) Returns the first non-NULL value in a list of expressions.

C) Converts a value from one data type to another.

D) Calculates the average of a set of values, ignoring NULLs.

Answer: B) Returns the first non-NULL value in a list of expressions.

Explanation: COALESCE(expr1, expr2, expr3, ...) evaluates the expressions in order and returns the current value of the first expression that does not evaluate to NULL.

---

95. What is a Foreign Key?

A) A key used to uniquely identify each row in a table.

B) A key that links two tables together.

C) A key that is composed of more than one column.

D) A key used for sorting data.

Answer: B) A key that links two tables together.

Explanation: A Foreign Key is a field (or collection of fields) in one table that uniquely identifies a row of another table. It is used to establish and enforce a link between the data in two tables. It must refer to a PRIMARY KEY or UNIQUE key in the other table.

---

96. The SQL AS keyword is used for what?

A) Filtering results.

B) Creating an alias for a table or column name.

C) Performing a comparison.

D) Adding a new record.

Answer: B) Creating an alias for a table or column name.

Explanation: An alias provides a temporary, more readable name for a column or table in a particular query. For example, SELECT FirstName AS Name FROM Employees;.

---

97. Which clause is processed last by the database engine in a SELECT statement?

A) FROM

B) WHERE

C) GROUP BY

D) ORDER BY

Answer: D) ORDER BY

Explanation: The logical processing order of a SELECT statement is generally: FROM -\> WHERE -\> GROUP BY -\> HAVING -\> SELECT -\> ORDER BY -\> LIMIT/OFFSET. The final sorting happens after all other operations are complete.

---

98. What is the result of SELECT NULL = NULL;?

A) TRUE

B) FALSE

C) NULL (or UNKNOWN)

D) Syntax Error

Answer: C) NULL (or UNKNOWN)

Explanation: In SQL, NULL represents an unknown value. You cannot test for NULL with comparison operators like = or \!=. Comparing an unknown value to another unknown value results in an

unknown outcome, which is represented by NULL. To check for NULL, you must use IS NULL or IS NOT NULL.

---

99. A SELF JOIN is:

A) A join that joins a table to another table.

B) A join that joins a table to itself.

C) A join that is not allowed in SQL.

D) Another name for an INNER JOIN.

Answer: B) A join that joins a table to itself.

Explanation: A self join is a regular join, but the table is joined with itself. This is useful for querying hierarchical data or comparing rows within the same table, such as finding employees who have the same manager.

---

100. A database transaction ends with:

A) END or STOP

B) COMMIT or ROLLBACK

C) SAVE or DISCARD

D) CLOSE or FINISH

Answer: B) COMMIT or ROLLBACK

Explanation: A transaction is a sequence of operations performed as a single logical unit of work. COMMIT makes all the data modifications performed during the transaction permanent. ROLLBACK undoes all the modifications made during the transaction.