

# Technology Questions and Answers

## Page 1

1. Which technology enables wireless charging for portable devices?  
Answer: c) Qi  
Explanation: Qi is a wireless charging standard that uses electromagnetic induction to transfer power to portable devices.
2. Which storage type is commonly used in modern laptops for faster performance?  
Answer: b) SSD  
Explanation: Solid State Drives (SSDs) offer faster read/write speeds compared to HDDs, making them the preferred choice for modern laptops.
3. Why is USB-C considered a universal standard for portable devices?  
Answer: c) It supports data transfer, power delivery, and video output  
Explanation: USB-C is versatile, supporting high-speed data transfer, power delivery (up to 100W), and video output (e.g., DisplayPort, HDMI), making it a universal connector.
4. What role does an eSIM play in modern smartphones and tablets?  
Answer: b) It replaces physical SIM cards with digital ones  
Explanation: An eSIM is an embedded SIM that allows users to switch carriers or plans digitally without a physical SIM card.
5. Arrange the following steps in the correct order to switch the laptop display between the built-in screen and an external monitor:  
Answer: 2, 4, 1, 3, 5  
Explanation:

1. Connect the external monitor to the laptop using the appropriate cable (e.g., HDMI, VGA).

2. Ensure the external monitor is powered on.

3. Press the function key (e.g., Fn) along with the designated display switch key (e.g., F4, F8, or similar).

4. Select the desired display mode (e.g., Duplicate, Extend, or Second Screen Only).

5. Verify the display configuration to ensure it works as expected.

## Page 2

6. Complete the table below by selecting the most appropriate cleaning method for each component from the options provided.

Laptop Component	Cleaning Procedure
1. Laptop Screen	b) Use a microfiber cloth dampened with a cleaning solution.
2. Keyboard	c) Use compressed air to blow out dirt.
3. Touchpad	a) Avoid excessive moisture and gently wipe the surface with an alcohol-based wipe or microfiber cloth.

- Explanation:

Laptop Screen:

 A microfiber cloth with a cleaning solution is safe for delicate screens.

Keyboard:

 Compressed air removes debris from between keys effectively.

Touchpad:

 Alcohol-based wipes or a microfiber cloth ensure safe cleaning without excessive moisture.
7. Compare the advantages and limitations of HDMI and VGA ports.  
Answer:

HDMI Advantages:

 Supports high-definition video (up to 4K), audio, and data over a single cable; compact connector; widely used in modern devices.

HDMI Limitations:

 Limited cable length (typically up to 15 meters); not backward compatible with VGA without adapters.

VGA Advantages:

 Widely compatible with older devices; longer cable lengths possible (up to 30 meters).

VGA Limitations:

 Analog signal limits resolution (typically up to 1920x1200); no audio support; bulkier connector.
8. Assess the significance of SIM card functionality in portable devices.  
Answer: SIM cards enable cellular connectivity, allowing devices to access mobile networks for calls, texts, and data. They are critical for communication, mobility, and accessing network services without relying on Wi-Fi. eSIMs enhance this by offering digital flexibility, reducing physical hardware needs.
9. Design a checklist for maintaining a personal laptop.  
Answer:

☐ Regularly update the operating system and software.

☐ Run antivirus scans to detect malware.

☐ Clean the keyboard, screen, and touchpad with appropriate methods.

☐ Back up important data to an external drive or cloud.

☐ Check storage space and remove unnecessary files.

☐ Ensure proper ventilation to prevent overheating.

☐ Update drivers for hardware components.

☐ Monitor battery health and avoid overcharging.
10. What is the primary difference between primitive and non-primitive data structures?  
Answer: b) Primitive structures store single data elements, while non-primitive structures store multiple related elements.  
Explanation: Primitive data structures (e.g., int, float) hold single values, while non-primitive structures (e.g., arrays, linked lists) store collections of related data.
11. What does a node in a linked list typically contain?  
Answer: a) Data and a pointer to the next node

**Explanation:** A node in a linked list contains data and a pointer to the next node, enabling dynamic linking of elements.

12. **Why is the choice of data structure important in programming?**

**Answer:** The choice of data structure impacts efficiency, memory usage, and performance. For example, arrays are fast for indexed access, while linked lists are better for dynamic insertions. Proper selection optimizes program execution and resource use.

13. **Why are linked lists preferred over arrays for dynamic data manipulation?**

**Answer:** Linked lists allow dynamic memory allocation, enabling efficient insertions and deletions without resizing. Arrays have fixed sizes, making resizing costly. Linked lists also support flexible data organization but may have slower access times.

14. **Write a C++ function to perform Binary Search on a sorted array.**

**Answer:**

```
int binarySearch(int arr[], int size, int target) {
    int left = 0, right = size - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (arr[mid] == target)
            return mid;
        if (arr[mid] < target)
            left = mid + 1;
        else
            right = mid - 1;
    }
    return -1; // Target not found
}
```

## Page 3

15. **Compare binary search and linear search algorithms.**

**Answer:**

- **Binary Search:** Requires a sorted array; has  $O(\log n)$  time complexity; efficient for large datasets; divides search space in half each step.
- **Linear Search:** Works on unsorted data; has  $O(n)$  time complexity; simpler but slower for large datasets; checks each element sequentially.

16. **Evaluate the Pre-order, In-order, and Post-order traversals of the given binary tree and select the correct option for each traversal.**

**Answer:**

- **Pre-order Traversal:** a) A, B, D, E, C, F (Root, Left, Right)
- **In-order Traversal:** c) B, D, E, A, F, C (Left, Root, Right)
- **Post-order Traversal:** d) D, E, B, F, C, A (Left, Right, Root)

17. **Design a natural language algorithm for implementing a circular queue using an array.**

**Answer:**

1. Initialize an array of fixed size, front and rear pointers to -1, and a size counter.
2. To enqueue:
  - If the queue is full (size equals array length), return an error.
  - If empty, set front to 0.
  - Increment rear (modulo array length), add the element, and increment size.
3. To dequeue:
  - If empty (size is 0), return an error.
  - Retrieve the element at front, increment front (modulo array length), and decrement size.
  - If size becomes 0, reset front and rear to -1.
4. To check if empty: Return true if size is 0.
5. To check if full: Return true if size equals array length.

18. **Which of the following is not a property of a computer network?**

**Answer:** c) Data destruction

**Explanation:** Scalability, reliability, and QoS are network properties, but data destruction is not a standard network characteristic.

19. **What does a node in a network represent?**

**Answer:** b) A computer or device connected to the network

**Explanation:** Nodes are devices (e.g., computers, printers) connected to a network for communication.

20. **Why is network scalability important?**

**Answer:** a) It allows adding more users without affecting performance.

**Explanation:** Scalability ensures a network can handle increased load efficiently.

## Page 4

20. **(continued)**

**Answer:** a) It allows adding more users without affecting performance.

**Explanation:** Scalability is critical for growth, not security, cost, or eliminating administrators.

21. **How does a router differ from a switch?**

**Answer:** b) A router connects multiple networks, while a switch connects devices within a LAN.

**Explanation:** Routers route traffic between networks (e.g., LAN to WAN), while switches connect devices within a single network.

22. **What is the role of the Access Point in a wireless network?**

**Answer:** c) To act as a central device for connecting wireless devices

**Explanation:** An Access Point enables wireless devices to connect to a network, acting as a bridge to wired networks.

23. **Demonstrate the process of setting up a peer-to-peer network.**

**Answer:**

1. Ensure all devices have network adapters.
2. Connect devices via Ethernet cables or Wi-Fi.
3. Assign IP addresses (manually or via DHCP).
4. Enable file and printer sharing in network settings.
5. Set up workgroup names (e.g., "WORKGROUP") on all devices.
6. Share resources (e.g., folders) and set permissions.
7. Test connectivity using ping or shared resource access.

24. **Analyze the following statements about firewalls in ensuring network security and determine whether each statement is True or False.**

**Answer:**

- a) True - Firewalls monitor and control traffic based on security rules.
- b) False - Firewalls reduce threats but cannot eliminate all cyber threats.
- c) True - Firewalls can be hardware, software, or both.
- d) False - Firewalls do not encrypt data; they filter traffic.
- e) False - Firewalls prevent unauthorized access, not increase it.

25. **Explain the role of DNS in a network environment.**

**Answer:** DNS (Domain Name System) translates domain names (e.g., google.com) into IP addresses, enabling devices to locate resources on the internet or local networks.

26. **Propose a design for a secure peer-to-peer file-sharing network.**

**Answer:**

- Use end-to-end encryption for file transfers.
- Implement user authentication with strong passwords or biometrics.
- Employ a decentralized architecture to avoid single points of failure.
- Use digital signatures to verify file integrity and authenticity.
- Include access control lists to restrict file access.
- Regularly update software to patch vulnerabilities.
- Log activities for monitoring and auditing.

27. **In a hierarchical model, data is organized in what structure?**

**Answer:** b) Tree

**Explanation:** Hierarchical models organize data in a tree-like structure with parent-child relationships.

28. **Why is data redundancy a problem in traditional file systems?**

**Answer:** b) It leads to storage wastage and inconsistencies.

**Explanation:** Redundancy wastes storage and can cause inconsistencies when data updates are not synchronized.

## Page 5

---

28. **(continued)**

**Answer:** b) It leads to storage wastage and inconsistencies.

**Explanation:** Redundancy does not enhance sharing or security; it complicates maintenance.

29. **Analyze the role of data independence in database architecture.**

**Answer:** Data independence allows changes to one level of a database (e.g., physical storage) without affecting others (e.g., logical schema). Physical independence enables storage optimization, while logical independence allows schema changes without impacting applications, improving flexibility and maintenance.

30. **Propose a method for implementing a backup and recovery strategy for a database.**

**Answer:**

- Schedule regular full and incremental backups.
- Store backups in multiple locations (e.g., local and cloud).
- Encrypt backup data for security.
- Test recovery procedures periodically.
- Use transaction logs for point-in-time recovery.
- Automate backups with tools like SQL Server Agent or cron jobs.
- Document the backup and recovery process.

31. **Which model is most commonly used at the conceptual level?**

**Answer:** d) Entity-Relationship Model

**Explanation:** The ER model is used to design databases conceptually, focusing on entities and relationships.

32. **What is the goal of database normalization?**

**Answer:** b) To eliminate redundancy and improve consistency

**Explanation:** Normalization organizes data to reduce redundancy and ensure data integrity.

33. **How does the logical level differ from the conceptual level?**

**Answer:** b) The logical level converts abstract concepts into relational structures, while the conceptual level defines the abstract framework.

**Explanation:** The conceptual level defines entities and relationships, while the logical level implements them as tables and keys.

34. **Why are referential columns used in relational databases?**

**Answer:** b) To link related data between tables

**Explanation:** Referential columns (foreign keys) link tables to maintain referential integrity.

35. **Match each step (Column A) with its correct description (Column B).**

**Answer:**

1. Identify and eliminate repeating groups → d) Separate multi-valued attributes into their own tables.
2. Remove partial dependencies → e) Ensure that every non-key attribute is fully dependent on the primary key.
3. Eliminate transitive dependencies → b) Ensure that no non-prime attribute depends on another non-prime attribute.
4. Organize data into distinct tables → a) Break down tables to ensure each table focuses on a single subject or concept.
5. Ensure the use of a primary key → c) Create a key column in every table to maintain data uniqueness and integrity.

## Page 6

36. Compare the features of conceptual, logical, and physical database design levels.

Answer:

- **Conceptual:** Defines entities, relationships, and attributes; high-level, independent of DBMS; focuses on user requirements.
- **Logical:** Translates conceptual design into relational schemas (tables, keys); DBMS-specific but independent of physical storage.
- **Physical:** Specifies storage structures, indexes, and file organization; optimizes performance and storage.

37. Evaluate the role of primary and foreign keys in maintaining referential integrity.

Answer: Primary keys uniquely identify records in a table, ensuring no duplicates. Foreign keys link related tables, enforcing referential integrity by ensuring values in the foreign key column exist in the referenced primary key. This maintains data consistency and prevents orphaned records.

38. Design an ERD for a library system with the following entities: Books, Members, and Transactions.

Answer:

- **Entities and Attributes:**
  - **Book:** ISBN (PK), authno, title, edition, category, price
  - **Reader:** UserId (PK), Email, address, phone no (multi-valued), name (firstname, lastname)
  - **Publisher:** PublisherId (PK), Year of publication, name
  - **Authentication System:** LoginId (PK), password
  - **Reports:** Reg\_no (PK), UserId, Book\_no, Issue/Return date
  - **Staff:** staff\_id (PK), name
  - **Reserve/Return:** Reserve date, Due date, Return date
- **Relationships:**
  - Reader reserves Books (1:N)
  - Publisher publishes Books (1:N)
  - Staff tracks Readers (M:N)
  - Staff maintains Reports (1:N)
  - Staff maintains Books (1:N)
  - Authentication provides login to Staff (1:N)

**ERD Description:**

- Boxes for each entity with attributes listed (PK underlined).
- Diamonds for relationships (e.g., "Reserves" between Reader and Book with 1:N notation).
- Lines connecting entities to relationships.
- Multi-valued attribute (phone no) in an oval with a double line.
- Composite attribute (name) split into firstname and lastname.

39. Which operator is used to obtain the address of a variable?

Answer: b) &

Explanation: The & operator returns the memory address of a variable in C++.

40. What does the dereference operator (\*) do in C++?

Answer: b) Access the value pointed to by a pointer

Explanation: The \* operator accesses the value stored at the address held by a pointer.

41. What is the difference between static and dynamic memory allocation?

Answer: a) Static memory is allocated at compile time, while dynamic memory is allocated at runtime.

Explanation: Static memory is fixed and allocated during compilation, while dynamic memory is allocated during execution using new/delete.

## Page 7

41. (continued)

Answer: a) Static memory is allocated at compile time, while dynamic memory is allocated at runtime.

Explanation: Static memory is faster, but dynamic memory is more flexible. Dynamic memory can be freed, unlike the incorrect option d.

42. Why are structures used in C++?

Answer: b) To group variables of different types under one name

Explanation: Structures organize related data of different types for easier management.

43. Complete the following C++ program by filling in the blanks to demonstrate the use of the new and delete operators.

Answer: The provided code is correct and complete.

```
#include <iostream>
using namespace std;
int main() {
    int *arr;
    arr = new int[5]; // Allocate memory for 5 elements
    for (int i = 0; i < 5; i++) {
        arr[i] = i + 1; // Assign values to the array
    }
    cout << "Array elements: ";
    for (int i = 0; i < 5; i++) {
        cout << arr[i] << " "; // Display the elements
    }
    cout << endl;
    delete[] arr; // Free memory
    return 0;
}
```

44. Write a C++ program to use a structure to store and display employee details (Name, ID, Salary).

Answer:

```

#include <iostream>
#include <string>
using namespace std;

struct Employee {
    string name;
    int id;
    float salary;
};

int main() {
    Employee emp;
    cout << "Enter name: ";
    getline(cin, emp.name);
    cout << "Enter ID: ";
    cin >> emp.id;
    cout << "Enter salary: ";
    cin >> emp.salary;

    cout << "\nEmployee Details:\n";
    cout << "Name: " << emp.name << endl;
    cout << "ID: " << emp.id << endl;
    cout << "Salary: " << emp.salary << endl;
    return 0;
}

```

45. **Analyze the advantages and risks of using dynamic memory allocation in C++.**

**Answer:**

- **Advantages:** Flexible memory use, allows runtime allocation, supports dynamic data structures (e.g., linked lists).
- **Risks:** Memory leaks if not deallocated, fragmentation, higher overhead than static allocation, potential for null pointer errors.

46. **Compare the use of pointers and arrays in C++ programs.**

**Answer:**

- **Pointers:** Store memory addresses; support dynamic allocation; flexible but error-prone (e.g., dangling pointers).
- **Arrays:** Fixed-size, contiguous memory; simpler to use; faster access but less flexible for resizing.

47. **Evaluate the efficiency of using structures for grouping data over separate variables.**

**Answer:** Structures improve code organization, readability, and maintainability by grouping related data. They reduce variable management overhead and enable passing multiple data items as a single unit, but they may slightly increase memory usage due to alignment.

48. **Design a C++ program to implement a singly linked list using structures and dynamic memory allocation to insert and display three elements 10, 20, and 30.**

**Answer:**

```

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node* next;
};

class LinkedList {
    Node* head;
public:
    LinkedList() : head(nullptr) {}

    void insert(int value) {
        Node* newNode = new Node;
        newNode->data = value;
        newNode->next = head;
        head = newNode;
    }

    void display() {
        Node* temp = head;
        while (temp) {
            cout << temp->data << " ";
            temp = temp->next;
        }
        cout << endl;
    }
};

int main() {
    LinkedList list;
    list.insert(30);
    list.insert(20);
    list.insert(10);
    cout << "Linked List: ";
    list.display();
    return 0;
}

```

## Page 8

49. Write a C++ program to demonstrate nested structures with a Student-Address relationship.  
**Answer:**

```

#include <iostream>
#include <string>
using namespace std;

struct Address {
    string province;
    string district;
};

struct Student {
    string name;
    int id;
    Address address;
    float marks;
    char grade;
};

int main() {
    Student stud;
    cout << "Enter name: ";
    getline(cin, stud.name);
    cout << "Enter ID: ";
    cin >> stud.id;
    cin.ignore();
    cout << "Enter province: ";
    getline(cin, stud.address.province);
    cout << "Enter district: ";
    getline(cin, stud.address.district);
    cout << "Enter marks: ";
    cin >> stud.marks;
    cout << "Enter grade: ";
    cin >> stud.grade;

    cout << "\nStudent Details:\n";
    cout << "Name: " << stud.name << endl;
    cout << "ID: " << stud.id << endl;
    cout << "Province: " << stud.address.province << endl;
    cout << "District: " << stud.address.district << endl;
    cout << "Marks: " << stud.marks << endl;
    cout << "Grade: " << stud.grade << endl;
    return 0;
}

```

50. What does encapsulation refer to in C++?

**Answer:** a) Binding data and functions together in a class

**Explanation:** Encapsulation combines data and methods into a class, restricting direct access to data for better control and security.

51. In C++, the symbol :: is called the \_\_\_\_\_ operator, used to define member functions outside a class.

**Answer:** Scope Resolution Operator

**Explanation:** The :: operator specifies the scope of a function or variable, often used to define class methods outside the class.

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

**Answer:** b) A non-member function with access to private data of a class

**Explanation:** A friend function is a non-member function declared as a friend in a class, allowing it to access private and protected members.

53. Complete the following statements about Object-Oriented Programming by filling in the blanks.

**Answer:**

- i. Object-Oriented Programming (OOP) emphasizes on **objects** rather than procedures.
- ii. A **class** is a blueprint for creating objects.
- iii. **Encapsulation** refers to combining data and functions into a single unit.
- iv. The concept of **inheritance** allows a class to inherit properties from another class.
- v. **Polymorphism** is used to define different behaviors for functions with the same name.

54. Which of the following demonstrates polymorphism in C++?

**Answer:** b) Operator overloading

**Explanation:** Polymorphism allows functions or operators to behave differently based on context, such as operator overloading.

55. True or False: In C++, the constructor of a base class is automatically inherited by derived class.

**Answer:** False

**Explanation:** Constructors are not inherited; derived classes must define their own or explicitly call the base class constructor.

56. Why is inheritance useful in C++?

**Answer:** b) It reduces code redundancy by reusing existing class code.

**Explanation:** Inheritance allows derived classes to reuse base class code, promoting modularity.

57. What is the role of a destructor in C++?

**Answer:** b) To release resources allocated to an object

**Explanation:** Destructors clean up resources (e.g., memory) when an object goes out of scope.

58. Match the following OOP concepts with their descriptions:

**Answer:**

- a) Encapsulation → 2) Hides implementation details from external access

- b) Polymorphism → 1) Allows multiple forms for a function
- c) Inheritance → 3) Enables reusing properties and methods from a base class

## Page 9

59. What is the significance of access specifiers in inheritance?

**Answer:** a) They dictate the visibility of base class members in derived classes.

**Explanation:** Access specifiers (public, protected, private) control how base class members are accessed in derived classes.

60. Write a C++ program to demonstrate single inheritance with base and derived classes for calculating the area of a rectangle.

**Answer:**

```
#include <iostream>
using namespace std;

class Shape {
protected:
    float length, width;
public:
    void setDimensions(float l, float w) {
        length = l;
        width = w;
    }
};

class Rectangle : public Shape {
public:
    float area() {
        return length * width;
    }
};

int main() {
    Rectangle rect;
    rect.setDimensions(5.0, 3.0);
    cout << "Area of rectangle: " << rect.area() << endl;
    return 0;
}
```

61. Write a C++ program using a friend function to access private data and calculate the tax on an employee's salary which is 30% of the salary.

**Answer:**

```
#include <iostream>
using namespace std;

class Employee {
    string name;
    float salary;
public:
    Employee(string n, float s) : name(n), salary(s) {}
    friend float calculateTax(Employee emp);
};

float calculateTax(Employee emp) {
    return 0.3 * emp.salary;
}

int main() {
    Employee emp("John", 50000);
    cout << "Tax: " << calculateTax(emp) << endl;
    return 0;
}
```

62. Write a program to demonstrate the use of a copy constructor in C++ for copying student's name and age details.

**Answer:**



```
#include <iostream>
#include <string>
using namespace std;

class Student {
    string name;
    int age;
public:
    Student(string n, int a) : name(n), age(a) {}
    Student(const Student &other) : name(other.name), age(other.age) {}
    void display() {
        cout << "Name: " << name << ", Age: " << age << endl;
    }
};

int main() {
    Student s1("Alice", 20);
    Student s2 = s1; // Copy constructor
    cout << "Original: ";
    s1.display();
    cout << "Copied: ";
    s2.display();
    return 0;
}
```

63. Compare the characteristics and applications of public and protected inheritance in C++.

**Answer:**

- **Public Inheritance:** Base class public and protected members remain public and protected in the derived class; used for "is-a" relationships (e.g., a Car is a Vehicle).
- **Protected Inheritance:** Base class public and protected members become protected in the derived class; used when the derived class needs restricted access to base class members.

64. Analyze the benefits of polymorphism in object-oriented design.

**Answer:** Polymorphism allows flexible and reusable code by enabling objects to be treated as instances of their base class. It simplifies maintenance, supports dynamic method resolution (via virtual functions), and enhances extensibility but may add runtime overhead.

65. Analyze the following statements about operator overloading and its effect on program readability.

**Answer:**

- a) True - Operator overloading makes code intuitive for familiar operations.
- b) True - Excessive overloading can confuse readers about operator behavior.
- c) True - Complex types may reduce clarity if overloaded operators are unfamiliar.
- d) True - Proper overloading clarifies complex operations.
- e) True - Overuse can lead to ambiguity in operator purpose.

## Page 10

65. (continued)

**Answer:** All statements (a-e) are True, as explained above.

66. Create a C++ program to demonstrate function overloading with different parameter types to add two numbers.

**Answer:**

```
#include <iostream>
using namespace std;

class Adder {
public:
    int add(int a, int b) {
        return a + b;
    }
    double add(double a, double b) {
        return a + b;
    }
};

int main() {
    Adder calc;
    cout << "Sum of integers: " << calc.add(5, 3) << endl;
    cout << "Sum of doubles: " << calc.add(5.5, 3.3) << endl;
    return 0;
}
```

67. Design a multi-level inheritance model in C++ for a library system that manages information about books, borrowed books, and library records.

**Answer:**

```

#include <iostream>
#include <string>
using namespace std;

class Book {
protected:
    string title;
public:
    void setTitle(string t) {
        title = t;
    }
};

class BorrowedBook : public Book {
protected:
    string borrower;
public:
    void setBorrower(string b) {
        borrower = b;
    }
};

class LibraryRecord : public BorrowedBook {
public:
    void display() {
        cout << "Book Title: " << title << endl;
        cout << "Borrower: " << borrower << endl;
    }
};

int main() {
    LibraryRecord record;
    record.setTitle("C++ Programming");
    record.setBorrower("Alice");
    record.display();
    return 0;
}

```

68. Create a C++ program to demonstrate operator overloading for a complex number class that takes two complex numbers and adds them.

**Answer:**

```

#include <iostream>
using namespace std;

class Complex {
    float real, imag;
public:
    Complex(float r = 0, float i = 0) : real(r), imag(i) {}
    Complex operator+(const Complex &other) {
        return Complex(real + other.real, imag + other.imag);
    }
    void display() {
        cout << real << " + " << imag << "i" << endl;
    }
};

int main() {
    Complex c1(3.5, 2.5), c2(1.5, 4.5);
    Complex sum = c1 + c2;
    cout << "Sum: ";
    sum.display();
    return 0;
}

```

69. What does Visual Basic stand for?

**Answer:** b) Visual Beginners' All-Purpose Symbolic Instruction Code

**Explanation:** Visual Basic is derived from BASIC, designed for easy application development.

70. Which version of Visual Basic introduced Rapid Application Development (RAD)?

**Answer:** c) Visual Basic 6.0

**Explanation:** VB 6.0 popularized RAD with its visual interface and drag-and-drop features.

71. What is a standard EXE project in Visual Basic used for?

**Answer:** b) Creating stand-alone applications

**Explanation:** Standard EXE projects create executable desktop applications in Visual Basic.

72. What are the key features of event-driven programming in Visual Basic?

**Answer:** b) Executes code based on user actions or events.

**Explanation:** Event-driven programming responds to user inputs (e.g., clicks) rather than sequential execution.

## Page 11

**73. What is the difference between an Option Button and a CheckBox in Visual Basic?**

**Answer:** c) Option Buttons allow single selection, while CheckBoxes allow multiple selections

**Explanation:** Option Buttons are mutually exclusive, while CheckBoxes allow multiple simultaneous selections.

**74. Write the steps to create a form in Visual Basic with a Label and a Button to display a message.**

**Answer:**

1. Open Visual Basic and create a new Standard EXE project.
2. In the Form Designer, drag a Label control from the Toolbox to the form.
3. Set the Label's Caption property to a desired text (e.g., "Welcome").
4. Drag a Button control to the form.
5. Set the Button's Caption property to "Click Me".
6. Double-click the Button to open the code editor.
7. Add code in the Button\_Click event, e.g., `MsgBox "Hello, World!"`.
8. Run the project to test the form.

**75. Compare the advantages and disadvantages of Visual Basic 6.0 for desktop application development.**

**Answer:**

- **Advantages:** Easy-to-use IDE, rapid development with drag-and-drop, supports event-driven programming, large community support.
- **Disadvantages:** Limited cross-platform support, outdated for modern applications, lacks advanced features like multithreading.

**76. Evaluate the differences between event-driven programming and procedural programming.**

**Answer:**

- **Event-Driven:** Code executes in response to events (e.g., user clicks); non-linear; ideal for GUI applications.
- **Procedural:** Code follows a sequential flow; structured around procedures; suited for algorithmic tasks.

**77. Design a Visual Basic program to simulate a login form with username and password validation.**

**Answer:**

```
Private Sub Form_Load()  
    ' Add two TextBoxes (Text1, Text2) and a CommandButton (Command1)  
    Text1.Text = "" ' Username  
    Text2.Text = "" ' Password  
    Command1.Caption = "Login"  
End Sub  
  
Private Sub Command1_Click()  
    If Text1.Text = "admin" And Text2.Text = "password123" Then  
        MsgBox "Login Successful!"  
    Else  
        MsgBox "Invalid Username or Password!"  
    End If  
End Sub
```

**78. Which of the following is a valid variable name in Visual Basic?**

**Answer:** None of the options are valid.

**Explanation:** Valid variable names cannot start with numbers (2Years), contain dots (My.house), spaces (Your Name), or special characters (Her&hermother). A valid example would be HerMother.

**79. Which data type is used to store a large range of numeric values in Visual Basic?**

**Answer:** c) Long

**Explanation:** Long supports a large range of integers (-2,147,483,648 to 2,147,483,647).

**80. What is the default data type assigned to variables not explicitly declared?**

**Answer:** b) Variant

**Explanation:** Variant is the default type in VB when a variable is not explicitly declared, allowing it to store any data type.

**81. What is the purpose of the Mod operator in Visual Basic?**

**Answer:** b) To find the remainder after division

**Explanation:** The Mod operator returns the remainder of a division operation (e.g.,  $10 \text{ Mod } 3 = 1$ ).

**82. What is the difference between numeric and non-numeric data types?**

**Answer:** b) Numeric data types can be manipulated mathematically, while non-numeric data types cannot.

**Explanation:** Numeric types (e.g., Integer, Double) support arithmetic, while non-numeric types (e.g., String) do not.

## Page 12

**83. Which logical operator in Visual Basic evaluates to True only when both operands are False?**

**Answer:** d) Nor

**Explanation:** The Nor operator returns True when both operands are False.

**84. How does a For...Next loop differ from a While loop?**

**Answer:** b) A For loop uses a counter variable, while a While loop uses a condition.

**Explanation:** For loops iterate a fixed number of times using a counter, while While loops iterate based on a condition.

**85. Write a Visual Basic program that uses a command button to calculate the sum and average of three floating-point numbers entered by the user through**

textboxes.

Answer:

```
Private Sub Form_Load()  
    Text1.Text = "" ' Number 1  
    Text2.Text = "" ' Number 2  
    Text3.Text = "" ' Number 3  
    Command1.Caption = "Calculate"  
End Sub  
  
Private Sub Command1_Click()  
    Dim num1 As Single, num2 As Single, num3 As Single  
    Dim sum As Single, avg As Single  
    num1 = CSng(Text1.Text)  
    num2 = CSng(Text2.Text)  
    num3 = CSng(Text3.Text)  
    sum = num1 + num2 + num3  
    avg = sum / 3  
    MsgBox "Sum: " & sum & vbCrLf & "Average: " & avg  
End Sub
```

86. Write a Visual Basic program to display on a listbox the multiplication table for a number entered by the user.

Answer:

```
Private Sub Form_Load()  
    Text1.Text = "" ' Input number  
    Command1.Caption = "Show Table"  
    List1.Clear  
End Sub  
  
Private Sub Command1_Click()  
    Dim num As Integer, i As Integer  
    List1.Clear  
    num = CInt(Text1.Text)  
    For i = 1 To 10  
        List1.AddItem num & " x " & i & " = " & num * i  
    Next i  
End Sub
```

87. Compare the use of If...Then...Else and Select Case structures in Visual Basic application decision-making.

Answer:

- **If...Then...Else:** Flexible for complex conditions; suitable for range-based or logical conditions; can be nested but may become complex.
- **Select Case:** Cleaner for multiple discrete conditions on a single variable; improves readability; less flexible for complex logic.

88. Analyze the benefits and drawbacks of using global variables in Visual Basic.

Answer:

- **Benefits:** Easy access across modules; simplifies data sharing.
- **Drawbacks:** Risk of unintended modifications; harder to debug; reduces modularity.

89. Evaluate the use of relational and logical operators in Visual Basic application decision-making.

**Answer:** Relational operators (e.g., =, <>) compare values, while logical operators (e.g., And, Or) combine conditions. Together, they enable complex decision-making, improving program logic and responsiveness but require careful use to avoid errors.

90. Design a Visual Basic program to simulate a grading system based on student marks using Select Case.

Answer:

```

Private Sub Form_Load()
    Text1.Text = "" ' Marks
    Command1.Caption = "Grade"
End Sub

Private Sub Command1_Click()
    Dim marks As Integer
    marks = CInt(Text1.Text)
    Select Case marks
        Case 90 To 100
            MsgBox "Grade: A"
        Case 80 To 89
            MsgBox "Grade: B"
        Case 70 To 79
            MsgBox "Grade: C"
        Case 60 To 69
            MsgBox "Grade: D"
        Case Else
            MsgBox "Grade: F"
    End Select
End Sub

```

91. Create a Visual Basic program to calculate the factorial of a number using a Do While loop.

**Answer:**

```

Private Sub Form_Load()
    Text1.Text = "" ' Input number
    Command1.Caption = "Calculate Factorial"
End Sub

Private Sub Command1_Click()
    Dim num As Long, fact As Long, i As Long
    num = CLng(Text1.Text)
    fact = 1
    i = 1
    Do While i <= num
        fact = fact * i
        i = i + 1
    Loop
    MsgBox "Factorial of " & num & " is " & fact
End Sub

```

92. What is the purpose of the 'static' keyword in Java?

**Answer:** b) To define shared class-level methods or variables

**Explanation:** Static members belong to the class, not instances, and are shared across all objects.

93. Which of the following is not a Java primitive data type?

**Answer:** None (all are primitive: Boolean, Float, Long, Byte)

**Explanation:** The question seems to expect a non-primitive type, but all options listed are primitive.

94. Why is Java considered platform-independent?

**Answer:** b) Java programs are compiled into bytecode that runs on any JVM.

**Explanation:** Bytecode is platform-agnostic and executed by the Java Virtual Machine.

## Page 13

94. (continued)

**Answer:** b) Java programs are compiled into bytecode that runs on any JVM.

**Explanation:** This enables Java's "write once, run anywhere" capability.

95. What happens when a Java program does not include a main method?

**Answer:** c) The JVM throws a runtime error.

**Explanation:** The JVM requires a main method as the entry point; its absence causes a `NoSuchMethodError`.

96. Write a Java program to print the multiplication table of a given number.

**Answer:**

```
import java.util.Scanner;

public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }
        scanner.close();
    }
}
```

97. Write a Java program to calculate the factorial of a number using a loop.

Answer:

```
import java.util.Scanner;

public class Factorial {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        long fact = 1;
        for (int i = 1; i <= num; i++) {
            fact *= i;
        }
        System.out.println("Factorial of " + num + " is " + fact);
        scanner.close();
    }
}
```

98. Compare the advantages and disadvantages of using loops versus recursion in Java based on the aspects below in the table:

Answer:

Aspect	Loops	Recursion
Performance	Faster; no call stack overhead	Slower due to repeated function calls
Memory Usage	Less memory; iterative	Higher memory; stack grows with calls
Readability	Clear for simple iterations	Elegant for recursive problems (e.g., trees)
Debugging	Easier; linear flow	Harder; stack trace complexity
Use Cases	Simple iterations (e.g., loops over arrays)	Complex problems (e.g., tree traversals)

99. Evaluate the advantages of Java's multithreading feature in modern applications.

Answer: Multithreading improves performance by parallel task execution, enhances responsiveness in GUI applications, and optimizes resource use in servers. However, it increases complexity, risks thread interference, and requires careful synchronization.

100. Write a Java program to demonstrate inheritance by creating a base class Vehicle with derived classes Car and Bike.

Answer:

```

class Vehicle {
    String brand;
    Vehicle(String brand) {
        this.brand = brand;
    }
    void display() {
        System.out.println("Brand: " + brand);
    }
}

class Car extends Vehicle {
    int wheels = 4;
    Car(String brand) {
        super(brand);
    }
    void show() {
        display();
        System.out.println("Wheels: " + wheels);
    }
}

class Bike extends Vehicle {
    int wheels = 2;
    Bike(String brand) {
        super(brand);
    }
    void show() {
        display();
        System.out.println("Wheels: " + wheels);
    }
}

public class InheritanceDemo {
    public static void main(String[] args) {
        Car car = new Car("Toyota");
        Bike bike = new Bike("Honda");
        car.show();
        bike.show();
    }
}

```

101. What is the primary purpose of the 'this' keyword in Java?

**Answer:** b) To refer to the current object

**Explanation:** The `this` keyword refers to the current instance, used to distinguish instance variables from local variables.

102. What type of inheritance does Java not support through classes?

**Answer:** c) Multiple

**Explanation:** Java supports single, hierarchical, and multilevel inheritance but not multiple inheritance through classes (only via interfaces).

103. What is the difference between default and parameterized constructors?

**Answer:** b) Parameterized constructors require arguments, while default constructors do not.

**Explanation:** Default constructors have no parameters and provide default values, while parameterized constructors initialize objects with provided values.

## Page 14

103. (continued)

**Answer:** b) Parameterized constructors require arguments, while default constructors do not.

**Explanation:** Default constructors are often compiler-generated if not defined explicitly.

104. What happens if a constructor is declared private?

**Answer:** c) Objects of the class cannot be created from outside the class.

**Explanation:** A private constructor restricts instantiation to within the class, often used for singletons.

105. Write a Java program to demonstrate the use of a default constructor and a parameterized constructor in a class called Person.

**Answer:**

```

```java
class Person {
    String name;
    int age;
    Person() {
        name = "Unknown";
        age = 0;
    }
    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    void displayInfo() {

```

```

        System.out.println("Name: " + name + ", Age: " + age);
    }
}
```

```

```

public class PersonDemo {
    public static void main(String[] args) {
        Person p1 = new Person();
        Person p2 = new Person("Alice", 25);
        p1.displayInfo();
        p2.displayInfo();
    }
}
...

```

106. Write a Java program to demonstrate the use of the `this` keyword for distinguishing local variables and instance variables in a class called `Student`.

**Answer:**

```

class Student {
    String name;
    int age;

    Student(String name, int age) {
        this.name = name; // Distinguish instance variable from parameter
        this.age = age;
    }

    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

public class StudentDemo {
    public static void main(String[] args) {
        Student s = new Student("Bob", 20);
        s.display();
    }
}

```

107. Compare the use of constructors and methods in Java. Highlight at least three key differences.

**Answer:**

- Constructors initialize objects and have no return type; methods perform actions and may return values.
- Constructors have the same name as the class; methods have distinct names.
- Constructors are called automatically during object creation; methods are called explicitly.

108. Examine the implications of method overriding in Java. Discuss its advantages and limitations.

**Answer:**

- **Advantages:** Enables polymorphism; allows customized behavior in derived classes; enhances flexibility.
- **Limitations:** Requires careful use of `@Override` to avoid errors; can increase complexity; may impact performance slightly.

109. Evaluate the importance of abstraction in designing Java applications.

**Answer:** Abstraction hides implementation details, simplifies complex systems, improves modularity, and enhances maintainability. It allows focusing on what an object does rather than how it does it, but over-abstraction can complicate design.

110. Design a Java program to demonstrate hierarchical inheritance where a base class `Animal` has a method `call()`, and two derived classes `Dog` and `Cat` inherit from it.

**Answer:**



```

class Animal {
    void call() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog barks");
    }
}

class Cat extends Animal {
    void meow() {
        System.out.println("Cat meows");
    }
}

public class HierarchicalInheritance {
    public static void main(String[] args) {
        Dog dog = new Dog();
        Cat cat = new Cat();
        dog.call();
        dog.bark();
        cat.call();
        cat.meow();
    }
}

```

111. What does the term "stream" refer to in Java IO?

**Answer:** b) A sequence of data values from an input source or output destination

**Explanation:** Streams represent a flow of data for reading from or writing to sources like files or consoles.

## Page 15

112. Which of the following methods is used to read a line of text in `BufferedReader`?

**Answer:** a) `readLine()`

**Explanation:** `BufferedReader`'s `readLine()` method reads a line of text from the input stream.

113. Why is the `Scanner` class preferred for reading user input?

**Answer:** b) It offers convenient methods for parsing primitive types and strings.

**Explanation:** `Scanner` provides methods like `nextInt()`, `nextDouble()`, and `nextLine()` for easy parsing.

114. How does the `DataInputStream` differ from `BufferedReader`?

**Answer:** a) `DataInputStream` reads byte-based data, while `BufferedReader` reads character-based data.

**Explanation:** `DataInputStream` handles binary data, while `BufferedReader` is optimized for text.

115. Write a Java program using `Scanner` to read an integer and a double from the user and display them.

**Answer:**

```

import java.util.Scanner;

public class ReadInput {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int num = scanner.nextInt();
        System.out.print("Enter a double: ");
        double val = scanner.nextDouble();
        System.out.println("Integer: " + num);
        System.out.println("Double: " + val);
        scanner.close();
    }
}

```

116. Write a Java program to copy content from one file to another using `FileInputStream` and `FileOutputStream`.

**Answer:**

```
import java.io.*;

public class FileCopy {
    public static void main(String[] args) {
        try {
            FileInputStream fis = new FileInputStream("input.txt");
            FileOutputStream fos = new FileOutputStream("output.txt");
            int byteData;
            while ((byteData = fis.read()) != -1) {
                fos.write(byteData);
            }
            fis.close();
            fos.close();
            System.out.println("File copied successfully");
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

117. Write a program to demonstrate reading a line of text from the console using `BufferedReader`.

**Answer:**

```
import java.io.*;

public class ReadConsole {
    public static void main(String[] args) {
        try {
            BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Enter text: ");
            String line = reader.readLine();
            System.out.println("You entered: " + line);
            reader.close();
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

118. Compare the advantages and disadvantages of `Scanner`, `BufferedReader`, and `DataInputStream`.

**Answer:**

- **Scanner:**
  - **Advantages:** Easy parsing of primitives and strings; user-friendly for console input.
  - **Disadvantages:** Slower for large data; not ideal for binary data.
- **BufferedReader:**
  - **Advantages:** Efficient for reading large text data; reduces I/O operations.
  - **Disadvantages:** Limited to character data; requires manual parsing.
- **DataInputStream:**
  - **Advantages:** Reads binary data (e.g., integers, floats) directly; efficient for structured data.
  - **Disadvantages:** Not suited for text; more complex for general input.

119. Critique the effectiveness of Java IO Streams for modern application development.

**Answer:**

- **Strengths:** Flexible for various data sources; supports both byte and character streams; robust for file and network I/O.
- **Weaknesses:** Complex API; can be slower than modern alternatives (e.g., NIO); requires manual resource management.

120. Design a Java program to append text to an existing file using `FileWriter`.

**Answer:**

```
import java.io.*;

public class AppendFile {
    public static void main(String[] args) {
        try {
            FileWriter fw = new FileWriter("output.txt", true); // Append mode
            fw.write("Appended text\n");
            fw.close();
            System.out.println("Text appended successfully");
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```