

---

## 1. Attempt any Six (Answer in one line) – *Expanded Answers*

### i) What is C++?

C++ is a **general-purpose, object-oriented programming language** developed by Bjarne Stroustrup that supports procedural, object-oriented, and generic programming, and is widely used for system software, application software, and game development.

---

### ii) What is a header file in C++?

A header file in C++ is a file with **.h extension** that contains **function declarations, macro definitions, constants, and class definitions**, which can be included in a program using the #include directive.

---

### iii) What is an array in C++?

An array in C++ is a **collection of elements of the same data type stored in contiguous memory locations**, which allows accessing elements using a common name and index values.

---

### iv) What is the difference between a structure and a union in C++?

In a structure, **each member has its own memory**, whereas in a union, **all members share the same memory location**, and only one member can store a value at a time.

---

### v) What is a friend function in C++?

A friend function is a **non-member function** that is allowed to **access private and protected members of a class** by using the friend keyword.

---

### vi) What is a recursive function in C++?

A recursive function is a function that **calls itself repeatedly** until a specified termination condition is met.

---

### vii) What is the syntax for switch statement in C++?

The switch statement evaluates an expression and executes the corresponding case block based on the matching value, with an optional default case.

---

### **viii) What is an inline function in C++?**

An inline function is a function that is **expanded at the point of function call**, reducing function call overhead and improving execution speed.

---

## **2. Attempt any Three – Long Answers**

### **i) Explain any two advantages of C++ programming language.**

#### **Advantage 1: Object-Oriented Programming (OOP)**

C++ supports object-oriented concepts such as **classes, objects, inheritance, polymorphism, encapsulation, and abstraction**, which help in organizing large programs, improving code reusability, and making maintenance easier.

#### **Advantage 2: High Performance and Efficiency**

C++ allows **low-level memory manipulation using pointers**, making it suitable for system programming, game development, and real-time applications where performance is critical.

---

### **ii) What are tokens in C++? Explain with examples.**

Tokens are the **smallest individual units of a C++ program** that are meaningful to the compiler.

#### **Types of Tokens in C++:**

1. Keywords (e.g., int, if, while)
2. Identifiers (e.g., variable names)
3. Constants (e.g., 10, 3.14)
4. Operators (e.g., +, -, \*)
5. Special symbols (e.g., {}, ;, ())

#### **Example:**

```
int sum = a + b;
```

Here, int, sum, =, a, +, b, ; are tokens.

---

### iii) What is the difference between a for loop and a do-while loop in C++?

#### For Loop

Entry-controlled loop

Condition checked before execution

May execute zero times

Suitable when iterations are known

#### Do-While Loop

Exit-controlled loop

Condition checked after execution

Executes at least once

Suitable when execution is required at least once

---

### iv) How do you declare a structure in C++?

A structure is declared using the struct keyword followed by member variables.

#### Example:

```
struct Student {  
    int rollNo;  
    char name[20];  
    float marks;  
};
```

This declaration creates a user-defined data type that can store different data types under one name.

---

### 3. Attempt any Two – Long Answers

#### i) Explain the concept of a pointer to a function in C++.

A function pointer is a **pointer that stores the address of a function**, allowing functions to be passed as arguments or called dynamically.

#### Syntax Example:

```
int add(int a, int b) {  
    return a + b;  
}
```

```
int (*ptr)(int, int);
```

```
ptr = add;
```

**Uses:**

- Dynamic function calling
  - Callback functions
  - Menu-driven programs
- 

**ii) Explain the use of memory management operators in C++.**

C++ provides memory management operators to **allocate and deallocate memory dynamically**.

**Operators:**

- new – Allocates memory at runtime
- delete – Frees allocated memory

**Example:**

```
int *p = new int;
```

```
delete p;
```

**Advantages:**

- Efficient memory usage
  - Prevents memory wastage
  - Useful for dynamic data structures
- 

**iii) What does the error directive do in C++?**

The #error directive is used to **generate a compilation error with a custom error message**, usually for debugging or conditional compilation.

**Example:**

```
#error Invalid input detected
```

---

**4. A) Attempt any One – 4 Marks****i) What is the use of the strlen() function in C++?**

The `strlen()` function is used to **calculate the length of a string**, excluding the null character `\0`.

**Example:**

```
char str[] = "Hello";  
  
cout << strlen(str);
```

**Output:** 5

---

**ii) What is pass by reference in C++?**

Pass by reference allows a function to **access and modify the actual variable** by passing its reference instead of a copy.

**Example:**

```
void update(int &x) {  
    x = x + 10;  
}
```

---

**4. B) Compulsory Question – 2 Marks**

**i) What is Relational operator?**

Relational operators are used to **compare two values** and return a boolean result (true or false).

**Examples:** `<`, `>`, `<=`, `>=`, `==`, `!=`

---

**5. Attempt any One – 6 Marks**

**i) Explain the difference between structure and union in C++.**

Structure	Union
Allocates separate memory for each member	Shares memory among all members
All members can store values simultaneously	Only one member can store value at a time
Size is sum of all members	Size is equal to largest member

## Structure

Safer and commonly used

## Union

Used for memory optimization

---

### ii) Explain the advantages of C++ over other programming languages.

1. Supports **multiple programming paradigms**
2. High performance and fast execution
3. Rich standard library (STL)
4. Portability across platforms
5. Better memory control using pointers
6. Widely used in real-world applications