

Institute Of Computer Science & Technology (I.C.S.T)

कम्प्यूटर विज्ञान एवं प्रौद्योगिकी संस्थान (आई. सी. एस. टी.)

We Create **Qualified**
IT Professionals



For More Info Visit



www.icstinstitute.in



Rakesh Marg, Ghaziabad



I.C.S.T®

INSTITUTE OF COMPUTER SCIENCE AND TECHNOLOGY

(An Autonomous Institution Registered Under the
Society & Public Trust Act-1882 by Govt. of India, NCT of Delhi.)

A Division of THE INDIAN EDUCATIONAL TRUST (Reg. No.308)

UP Govt. & NCT of Delhi | MSME
DOEACC (NIELIT) For CCC
ISO 21001:2018 Certified Institution

ADCA

DCA

DFA

BASIC

CCC

BCC

Web Designing

Tally GST

Python

JAVA | .net

C/C++

SQL

For Any Query Contact Us On

9650571128 | 9910863033

SAMPLE QUESTION ANSWER FOR THEORY EXAMINATION

&

Effective Preparation for Various Interview Questions

For

PROGRAMMING LANGUAGES

Prepared by :- **Abhishek Sir**

PROGRAMMING LANGUAGES(C,C++)

Sample Question Answer for MCQ Examination

Q1. What is the C Programming Language? (10 Marks)

English Answer

Introduction

C is a high-level, general-purpose, procedural programming language developed by **Dennis Ritchie** in **1972** at **Bell Laboratories**. It was originally designed to develop the UNIX operating system, but due to its speed, efficiency, and portability, it became one of the most widely used programming languages in the world. C is often called the "**Mother of Programming Languages**" because many modern languages such as C++, Java, C#, and Python are influenced by it.

Definition

C is a structured programming language that allows programmers to write efficient, fast, and portable programs. It provides low-level memory access through pointers while also supporting high-level programming constructs. It is suitable for developing both system software and application software.

Features of C Programming

1. **Simple and Easy to Learn** – C has a small set of keywords and a simple syntax.
2. **Structured Programming** – Programs are divided into functions, making them easy to understand and maintain.
3. **Portable** – C programs can run on different operating systems with little or no modification.
4. **Fast Execution** – Since C is a compiled language, programs execute quickly.
5. **Efficient Memory Management** – Pointers allow direct access to memory.
6. **Rich Library Functions** – Standard libraries provide many built-in functions.
7. **Modularity** – Large programs can be divided into smaller functions.
8. **Extensible** – Users can create their own functions and libraries.

Applications of C

- Operating System Development
- Compiler Design
- Embedded Systems
- Device Drivers
- Database Systems
- Networking Software

- Scientific Applications
- Game Development

Advantages of C

- High performance
- Easy to debug
- Portable across platforms
- Supports structured programming
- Efficient memory utilization
- Foundation for learning other programming languages

Limitations of C

- Does not support Object-Oriented Programming.
- No automatic garbage collection.
- Less secure because pointers can be misused.
- No exception handling.

Example

```
#include<stdio.h>

int main()
{
    printf("Hello, World!");
    return 0;
}
```

Output

Hello, World!

Conclusion

C is one of the most important programming languages in computer science. Its simplicity, efficiency, portability, and speed make it suitable for developing operating systems, embedded systems, and application software. Learning C builds a strong foundation for understanding advanced programming languages.

प्रश्न 1. C Programming Language क्या है? (10 अंक)

हिंदी उत्तर

परिचय

C एक उच्च स्तरीय (High-Level), सामान्य प्रयोजन (General Purpose) तथा प्रक्रियात्मक (Procedural) प्रोग्रामिंग भाषा है। इसका विकास **डेनिस रिची (Dennis Ritchie)** ने 1972 में **Bell Laboratories** में किया था। इसे प्रारम्भ में UNIX ऑपरेटिंग सिस्टम विकसित करने के लिए बनाया गया था, लेकिन इसकी गति, दक्षता तथा पोर्टेबिलिटी के कारण यह दुनिया की सबसे लोकप्रिय प्रोग्रामिंग भाषाओं में से एक बन गई। C को "**Programming Languages की जननी (Mother of Programming Languages)**" भी कहा जाता है क्योंकि C++, Java, C#, और Python जैसी कई आधुनिक भाषाएँ इससे प्रभावित हैं।

परिभाषा

C एक संरचित (Structured) प्रोग्रामिंग भाषा है जो तेज़, कुशल तथा पोर्टेबल प्रोग्राम बनाने की सुविधा प्रदान करती है। इसमें पॉइंटर (Pointer) के माध्यम से मेमोरी तक सीधी पहुँच संभव होती है। इसका उपयोग सिस्टम सॉफ्टवेयर और एप्लिकेशन सॉफ्टवेयर दोनों के विकास में किया जाता है।

C भाषा की विशेषताएँ

1. **सरल एवं सीखने में आसान** – इसमें सीमित Keywords और आसान Syntax होता है।
2. **Structured Programming** – प्रोग्राम को छोटे-छोटे Functions में विभाजित किया जा सकता है।
3. **Portable** – एक प्लेटफॉर्म पर लिखा गया प्रोग्राम दूसरे प्लेटफॉर्म पर भी चलाया जा सकता है।
4. **तेज़ निष्पादन (Fast Execution)** – Compiler के कारण C प्रोग्राम बहुत तेज़ चलते हैं।
5. **Memory Management** – Pointer के माध्यम से मेमोरी का प्रभावी उपयोग किया जा सकता है।
6. **Rich Library** – अनेक Standard Library Functions उपलब्ध हैं।
7. **Modularity** – बड़े प्रोग्राम को छोटे Modules में बाँटा जा सकता है।
8. **User-defined Functions** – उपयोगकर्ता अपनी आवश्यकता के अनुसार नए Functions बना सकता है।

C भाषा के उपयोग

- ऑपरेटिंग सिस्टम बनाना
- कम्पाइलर विकसित करना
- एम्बेडेड सिस्टम
- डिवाइस ड्राइवर
- डेटाबेस सॉफ्टवेयर
- नेटवर्किंग प्रोग्राम
- वैज्ञानिक अनुप्रयोग
- गेम डेवलपमेंट

लाभ (Advantages)

- तेज़ और प्रभावी
- पोर्टेबल
- मेमोरी का कुशल उपयोग
- डिबग करना आसान
- अन्य भाषाओं की मजबूत नींव

सीमाएँ (Limitations)

- Object-Oriented Programming का समर्थन नहीं करती।
- Automatic Garbage Collection नहीं है।
- Pointer के गलत उपयोग से त्रुटियाँ हो सकती हैं।
- Exception Handling उपलब्ध नहीं है।

उदाहरण

```
#include<stdio.h>

int main()
{
    printf("Hello, World!");
    return 0;
}
```

आउटपुट

Hello, World!

निष्कर्ष

C प्रोग्रामिंग भाषा कंप्यूटर विज्ञान की सबसे महत्वपूर्ण भाषाओं में से एक है। इसकी सरलता, गति, पोर्टेबिलिटी और दक्षता के कारण इसका उपयोग आज भी ऑपरेटिंग सिस्टम, एम्बेडेड सिस्टम, कम्पाइलर और अनेक प्रकार के सॉफ्टवेयर विकसित करने में किया जाता है। C सीखने से अन्य आधुनिक प्रोग्रामिंग भाषाओं को समझना आसान हो जाता है।

Q2. What is the C++ Programming Language? (10 Marks)

English Answer

Introduction

C++ is a powerful, general-purpose programming language developed by **Bjarne Stroustrup** at **Bell Laboratories** in **1983**. It is an extension of the C programming language and was designed to add **Object-Oriented Programming (OOP)** features while retaining the speed and efficiency of C. C++ is one of the most widely used programming languages for developing operating systems, desktop applications, games, embedded systems, compilers, and high-performance software.

Definition

C++ is a **general-purpose, compiled programming language** that supports **procedural programming, object-oriented programming (OOP), and generic programming**. It allows programmers to create efficient, reusable, and maintainable software by using concepts such as classes, objects, inheritance, polymorphism, encapsulation, and abstraction.

History of C++

- Developed by **Bjarne Stroustrup** in **1983**.
- Originally called "**C with Classes**."
- Created to overcome the limitations of the C language.
- Standardized by the **International Organization for Standardization (ISO)**.
- Modern versions include **C++11, C++14, C++17, C++20, and C++23**, each introducing new features.

Features of C++

1. **Object-Oriented Programming (OOP)** – Supports classes and objects.
2. **Platform Independent** – Programs can run on multiple operating systems after compilation.
3. **Fast Execution** – Compiled language with high performance.
4. **Inheritance** – Enables code reuse.
5. **Polymorphism** – Allows one interface to perform multiple functions.
6. **Encapsulation** – Protects data by combining data and functions into a single unit.
7. **Abstraction** – Hides unnecessary implementation details.
8. **Function Overloading** – Multiple functions can have the same name with different parameters.
9. **Operator Overloading** – Operators can be redefined for user-defined data types.
10. **Exception Handling** – Helps manage runtime errors efficiently.

11. **Templates** – Support generic programming.
12. **Standard Template Library (STL)** – Provides ready-made data structures and algorithms.

Applications of C++

- Operating System Development
- Game Development
- Software Applications
- Web Browsers
- Database Management Systems
- Embedded Systems
- Device Drivers
- Computer Graphics
- Artificial Intelligence
- Financial Software

Advantages of C++

- High performance and speed.
- Supports both procedural and object-oriented programming.
- Code reusability through inheritance.
- Better security using encapsulation.
- Efficient memory management.
- Rich Standard Library (STL).
- Suitable for large-scale software development.
- Portable and reliable.

Limitations of C++

- Complex syntax for beginners.
- Manual memory management may cause memory leaks.
- No automatic garbage collection.
- Large programs can become difficult to maintain without proper design.
- Compilation time may be longer than some modern languages.

Example Program

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

```
class Student
{
```



```
public:
    void display()
    {
        cout << "Welcome to C++ Programming";
    }
};

int main()
{
    Student s;
    s.display();
    return 0;
}
```

Output

Welcome to C++ Programming

Conclusion

C++ is one of the most powerful and versatile programming languages available today. It combines the efficiency of C with advanced object-oriented programming features, making it suitable for developing everything from simple applications to complex operating systems and games. Learning C++ provides a strong foundation for understanding software development, data structures, algorithms, and modern programming concepts.

प्रश्न 2. C++ Programming Language क्या है? (10 अंक)

हिंदी उत्तर

परिचय

C++ एक शक्तिशाली (Powerful), सामान्य प्रयोजन (General Purpose) प्रोग्रामिंग भाषा है जिसका विकास **ब्यार्ने स्ट्रॉस्ट्रूप (Bjarne Stroustrup)** ने 1983 में Bell Laboratories में किया था। यह C भाषा का विस्तारित (Extension) रूप है, जिसमें **Object-Oriented Programming (OOP)** की विशेषताएँ जोड़ी गई हैं। C++ अपनी तेज गति, उच्च प्रदर्शन तथा पुनः उपयोग योग्य (Reusable) कोड लिखने की क्षमता के कारण विश्व की सबसे लोकप्रिय प्रोग्रामिंग भाषाओं में से एक है।

परिभाषा

C++ एक **Compiled Programming Language** है जो **Procedural Programming, Object-Oriented Programming (OOP)** तथा **Generic Programming** का समर्थन करती है। इसमें **Class, Object, Inheritance,**

Polymorphism, Encapsulation तथा Abstraction जैसी आधुनिक विशेषताएँ उपलब्ध हैं, जिनकी सहायता से बड़े और जटिल सॉफ्टवेयर आसानी से विकसित किए जा सकते हैं।

C++ का इतिहास

- C++ का विकास **Bjarne Stroustrup** ने 1983 में किया।
- प्रारम्भ में इसका नाम "**C with Classes**" था।
- इसे C भाषा की सीमाओं को दूर करने के लिए बनाया गया।
- बाद में ISO (International Organization for Standardization) द्वारा इसे मानकीकृत किया गया।
- वर्तमान में C++11, C++14, C++17, C++20 तथा C++23 इसके आधुनिक संस्करण हैं।

C++ की विशेषताएँ

1. **Object-Oriented Programming** – Class और Object पर आधारित।
2. **Platform Independent** – विभिन्न ऑपरेटिंग सिस्टम पर कार्य कर सकती है।
3. **High Speed** – Compiler के कारण प्रोग्राम तेज़ी से चलते हैं।
4. **Inheritance** – कोड का पुनः उपयोग संभव बनाता है।
5. **Polymorphism** – एक ही Function कई रूपों में कार्य कर सकता है।
6. **Encapsulation** – डेटा को सुरक्षित रखता है।
7. **Abstraction** – आवश्यक जानकारी दिखाता है और जटिलता छिपाता है।
8. **Function Overloading** – एक ही नाम के कई Functions बनाए जा सकते हैं।
9. **Operator Overloading** – Operators को अपनी आवश्यकता के अनुसार परिभाषित किया जा सकता है।
10. **Exception Handling** – Runtime Errors को संभालने की सुविधा।
11. **Templates** – Generic Programming का समर्थन।
12. **Standard Template Library (STL)** – अनेक तैयार Data Structures और Algorithms उपलब्ध कराती है।

C++ के उपयोग

- ऑपरेटिंग सिस्टम
- गेम डेवलपमेंट
- डेस्कटॉप सॉफ्टवेयर
- वेब ब्राउज़र
- डेटाबेस मैनेजमेंट सिस्टम
- एम्बेडेड सिस्टम
- डिवाइस ड्राइवर
- कंप्यूटर ग्राफिक्स
- आर्टिफिशियल इंटेलिजेंस
- बैंकिंग एवं फाइनेंशियल सॉफ्टवेयर

C++ के लाभ (Advantages)

- तेज़ और प्रभावी।
- Procedural तथा Object-Oriented दोनों प्रकार की प्रोग्रामिंग का समर्थन।
- Inheritance द्वारा Code Reusability।
- Encapsulation द्वारा Data Security।
- बेहतर Memory Management।
- STL के कारण विकास कार्य आसान।
- बड़े Software Projects के लिए उपयुक्त।
- Portable तथा Reliable।

C++ की सीमाएँ (Limitations)

- शुरुआती विद्यार्थियों के लिए सीखना थोड़ा कठिन।
- Memory Management स्वयं करना पड़ता है।
- Automatic Garbage Collection नहीं है।
- बड़े प्रोग्रामों का रखरखाव कठिन हो सकता है।
- Compilation में अधिक समय लग सकता है।

उदाहरण (Example Program)

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

```
class Student
{
public:
    void display()
    {
        cout << "Welcome to C++ Programming";
    }
};
```

```
int main()
```

```
{
    Student s;
    s.display();
    return 0;
}
```

आउटपुट

Welcome to C++ Programming

निष्कर्ष

C++ एक आधुनिक, शक्तिशाली और बहुउद्देशीय प्रोग्रामिंग भाषा है। यह C की गति और दक्षता के साथ Object-Oriented Programming की उन्नत सुविधाएँ प्रदान करती है। इसका उपयोग ऑपरेटिंग सिस्टम, गेम, डेटाबेस, एम्बेडेड सिस्टम और बड़े सॉफ्टवेयर अनुप्रयोगों के विकास में किया जाता है। इसलिए C++ सीखना किसी भी कंप्यूटर विज्ञान के विद्यार्थी के लिए अत्यंत महत्वपूर्ण है।

Q3. What are the Basic Data Types in C and C++? (10 Marks)

English Answer

Introduction

Data is the most important part of any computer program. Every program stores and processes different kinds of data such as numbers, characters, and decimal values. To store these values correctly, C and C++ provide **data types**. A data type specifies the type of value a variable can store, the amount of memory required, and the range of values it can hold. Choosing the correct data type helps improve memory usage and program performance.

Definition

A **data type** is a keyword that tells the compiler what type of data a variable will store. It determines the size of memory allocated to the variable and the operations that can be performed on it.

Example:

```
int age = 20;  
float salary = 25000.50;  
char grade = 'A';
```

In the above example:

- **int** stores an integer value.
- **float** stores a decimal value.
- **char** stores a single character.

Basic Data Types in C and C++

1. int (Integer)

- Used to store whole numbers.
- Memory: Usually **4 bytes**
- Example:

```
int marks = 95;
```

2. float

- Used to store decimal numbers.
- Memory: Usually **4 bytes**
- Precision: About **6–7 decimal digits**

Example:

```
float price = 99.75;
```

3. double

- Used for large decimal numbers.
- Memory: Usually **8 bytes**
- More accurate than float.

Example:

```
double pi = 3.1415926535;
```

4. char

- Used to store a single character.
- Memory: **1 byte**

Example:

```
char grade = 'A';
```

5. void

- Represents **no value**.
- Mainly used in functions that do not return any value.

Example:

```
void display()  
{
```

```
cout<<"Hello";  
}
```

6. bool (Only in C++)

- Stores logical values.
- Possible values are **true** or **false**.
- Memory: Usually **1 byte**

Example:

```
bool result = true;
```

Data Type Modifiers

Modifiers change the size or range of a data type.

- **short**
- **long**
- **signed**
- **unsigned**

Example:

```
long int population;  
unsigned int age;
```

User-Defined Data Types

C++ also provides user-defined data types.

- Structure (struct)
- Union
- Enumeration (enum)
- Class (C++ only)

Example:

```
class Student  
{  
public:  
    int roll;  
    char name[20];  
};
```


Advantages of Data Types

1. Efficient memory utilization.
2. Improves program performance.
3. Reduces programming errors.
4. Makes code easier to understand.
5. Helps the compiler detect type-related mistakes.
6. Ensures proper storage of data.

Limitations

- Every data type has a fixed memory size.
- Wrong data type selection may waste memory.
- Integer types cannot store decimal values.
- Character type stores only one character.

Summary Table

Data Type	Purpose	Typical Size
-----------	---------	--------------

int	Integer values	4 Bytes
-----	----------------	---------

float	Decimal values	4 Bytes
-------	----------------	---------

double	Large decimal values	8 Bytes
--------	----------------------	---------

char	Single character	1 Byte
------	------------------	--------

void	No value	0 Byte
------	----------	--------

bool (C++)	True/False	1 Byte
------------	------------	--------

Conclusion

Data types are the foundation of programming in C and C++. They define what type of information a variable can store and how much memory it requires. Proper selection of data types improves program efficiency, reduces memory usage, and makes software more reliable. Therefore, understanding data types is essential for every C and C++ programmer.

प्रश्न 3. C और C++ में Basic Data Types क्या हैं? (10 अंक)

हिंदी उत्तर

परिचय

किसी भी कंप्यूटर प्रोग्राम का सबसे महत्वपूर्ण भाग **डेटा (Data)** होता है। प्रोग्राम विभिन्न प्रकार के डेटा जैसे संख्या (Numbers), दशमलव (Decimal Values) और अक्षर (Characters) को संग्रहित और संसाधित करता है। इन सभी प्रकार के डेटा को सही तरीके से संग्रहित करने के लिए C और C++ में **Data Types** का उपयोग किया जाता है। Data Type यह निर्धारित करता है कि कोई वैरिएबल किस प्रकार का डेटा रखेगा, उसे कितनी मेमोरी मिलेगी तथा वह कितनी सीमा (Range) तक मान संग्रहित कर सकता है।

परिभाषा

Data Type वह कीवर्ड (Keyword) है जो कम्पाइलर को बताता है कि किसी वैरिएबल में किस प्रकार का डेटा संग्रहित किया जाएगा। यह मेमोरी का आकार (Memory Size), मानों की सीमा (Range) तथा उस पर किए जाने वाले ऑपरेशन को निर्धारित करता है।

उदाहरण:

```
int age = 20;  
float salary = 25000.50;  
char grade = 'A';
```

यहाँ:

- **int** पूर्णांक (Integer) संग्रहित करता है।
- **float** दशमलव संख्या संग्रहित करता है।
- **char** एक अक्षर संग्रहित करता है।

C और C++ के मुख्य Data Types

1. int (Integer)

- पूर्णांक (Whole Numbers) संग्रहित करने के लिए।
- सामान्यतः **4 Bytes** मेमोरी।

उदाहरण:

```
int marks = 95;
```

2. float

- दशमलव संख्या (Decimal Number) संग्रहित करने के लिए।
- सामान्यतः **4 Bytes** मेमोरी।
- लगभग 6–7 दशमलव अंकों तक सटीकता।

उदाहरण:

float price = 99.75;

3. double

- बड़े दशमलव मानों के लिए।
- सामान्यतः **8 Bytes** मेमोरी।
- float की तुलना में अधिक सटीक।

उदाहरण:

double pi = 3.1415926535;

4. char

- एक अक्षर (Single Character) संग्रहित करने के लिए।
- सामान्यतः **1 Byte** मेमोरी।

उदाहरण:

char grade = 'A';

5. void

- किसी भी प्रकार का मान (No Value) प्रदर्शित नहीं करता।
- उन Functions में उपयोग होता है जो कोई मान वापस नहीं करते।

उदाहरण:

```
void display()  
{  
    cout<<"Hello";  
}
```


6. bool (केवल C++ में)

- Logical मान संग्रहित करता है।
- केवल दो मान होते हैं: **true** या **false**।
- सामान्यतः **1 Byte** मेमोरी।

उदाहरण:

```
bool result = true;
```

Data Type Modifiers

Modifiers का उपयोग Data Type के आकार (Size) या सीमा (Range) को बदलने के लिए किया जाता है।

- short
- long
- signed
- unsigned

उदाहरण:

```
long int population;  
unsigned int age;
```

User-Defined Data Types

C++ में उपयोगकर्ता स्वयं भी Data Types बना सकता है।

- Structure (struct)
- Union
- Enumeration (enum)
- Class

उदाहरण:

```
class Student  
{  
public:  
    int roll;  
    char name[20];  
};
```

Data Types के लाभ

1. मेमोरी का सही उपयोग होता है।
2. प्रोग्राम की गति बढ़ती है।
3. त्रुटियाँ कम होती हैं।
4. कोड पढ़ना और समझना आसान होता है।
5. Compiler Type Errors को पहचान सकता है।
6. डेटा सुरक्षित और सही रूप में संग्रहित होता है।

सीमाएँ (Limitations)

- प्रत्येक Data Type का निश्चित आकार होता है।
- गलत Data Type चुनने पर मेमोरी व्यर्थ हो सकती है।
- Integer Data Type दशमलव संख्या संग्रहित नहीं कर सकता।
- Char केवल एक अक्षर संग्रहित करता है।

सारणी (Summary Table)

Data Type उपयोग सामान्य आकार

int पूर्णांक 4 Bytes

float दशमलव संख्या 4 Bytes

double बड़े दशमलव मान 8 Bytes

char एक अक्षर 1 Byte

void कोई मान नहीं 0 Byte

bool (C++) True/False 1 Byte

निष्कर्ष

Data Types, C और C++ प्रोग्रामिंग की आधारशिला हैं। ये निर्धारित करते हैं कि कोई वेरिएबल किस प्रकार का डेटा संग्रहित करेगा, उसे कितनी मेमोरी मिलेगी तथा उस पर कौन-से ऑपरेशन किए जा सकते हैं। सही Data Type का चयन करने से प्रोग्राम तेज़, विश्वसनीय और मेमोरी-कुशल बनता है। इसलिए प्रत्येक प्रोग्रामर के लिए Data Types की अच्छी समझ होना अत्यंत आवश्यक है।

Q4. What is the difference between C and C++? (Short Answer)

English:

C is a **procedural programming language** that focuses on functions and procedures, whereas C++ is an **object-oriented programming language** that supports classes and objects. C++ includes features such as **inheritance, polymorphism, encapsulation, function overloading, and exception handling**, which are not available in C. C is mainly used for system programming, while C++ is suitable for both system and application development.

हिंदी:

C एक **Procedural Programming Language** है जो Functions पर आधारित होती है, जबकि C++ एक **Object-Oriented Programming Language** है जो Class और Object पर आधारित होती है। C++ में **Inheritance, Polymorphism, Encapsulation, Function Overloading** तथा **Exception Handling** जैसी सुविधाएँ उपलब्ध हैं, जो C में नहीं होतीं। C का उपयोग मुख्यतः सिस्टम प्रोग्रामिंग के लिए तथा C++ का उपयोग सिस्टम और एप्लिकेशन दोनों के विकास के लिए किया जाता है।

Q5. What is a Pointer in C and C++? (Short Answer)

English:

A **pointer** is a variable that stores the **memory address** of another variable. It is declared using the **(*)** symbol. Pointers are used for dynamic memory allocation, passing arguments to functions, and efficient memory management. In C++, smart pointers provide safer memory management.

हिंदी:

Pointer एक ऐसा Variable होता है जो किसी अन्य Variable का **Memory Address** संग्रहित करता है। इसे **(*)** चिन्ह द्वारा घोषित किया जाता है। Pointer का उपयोग Dynamic Memory Allocation, Function में Address Pass करने तथा Memory Management के लिए किया जाता है। C++ में **Smart Pointer** अधिक सुरक्षित Memory Management प्रदान करते हैं।

Q6. What is a Variable in C and C++? (Short Answer)

English:

A **variable** is a named memory location used to store data. Every variable must be declared with a data type before use. Its value can change during program execution. Variables may be **local, global, static, or dynamic**, depending on their scope and lifetime.

हिंदी:

Variable मेमोरी में एक **नामित स्थान (Named Memory Location)** होता है, जिसका उपयोग डेटा संग्रहित करने के लिए किया जाता है। उपयोग से पहले प्रत्येक Variable को किसी **Data Type** के साथ घोषित करना आवश्यक होता है। प्रोग्राम के दौरान इसका मान बदला जा सकता है। Scope और Lifetime के आधार पर Variable **Local, Global, Static** तथा **Dynamic** प्रकार के हो सकते हैं।

Q7. What are Functions in C and C++? (10 Marks)

English Answer

Introduction

A function is one of the most important concepts in C and C++ programming. Large programs become easier to understand and maintain when they are divided into smaller parts called functions. Each function performs a specific task and can be called whenever needed.

Definition

A **function** is a named block of reusable code that performs a specific task. It may receive input through parameters and can return a value to the calling function.

Syntax

```
return_type function_name(parameter_list)
{
    // Statements
}
```

Components of a Function

1. Function Declaration (Prototype)
2. Function Definition
3. Function Call
4. Return Statement

Types of Functions

1. Library Functions

These are predefined functions provided by the compiler.

Examples:

- printf()
- scanf()
- sqrt()
- strlen()

2. User-Defined Functions

These are created by the programmer according to program requirements.

Advantages of Functions

- Code Reusability
- Reduces Program Size
- Easy Debugging
- Better Readability
- Modular Programming
- Easy Maintenance

Example

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

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

```
}  
  
int main()  
{  
    cout<<"Sum = "<<add(10,20);  
    return 0;  
}
```

Output

Sum = 30

Conclusion

Functions make programs modular, reusable, and easier to understand. They reduce repetition and improve program efficiency.

हिंदी उत्तर

परिचय

Function, C और C++ की सबसे महत्वपूर्ण अवधारणाओं में से एक है। बड़े प्रोग्राम को छोटे-छोटे भागों में विभाजित करने के लिए Function का उपयोग किया जाता है। प्रत्येक Function एक विशेष कार्य करता है और आवश्यकता पड़ने पर कई बार उपयोग किया जा सकता है।

परिभाषा

Function एक नामित (Named) कोड ब्लॉक होता है, जो किसी विशेष कार्य को करता है। यह Parameters के माध्यम से इनपुट प्राप्त कर सकता है तथा Return Statement द्वारा परिणाम वापस कर सकता है।

Function के भाग

1. Function Declaration
2. Function Definition
3. Function Call
4. Return Statement

Function के प्रकार

1. Library Function

- printf()
- scanf()
- sqrt()
- strlen()

2. User Defined Function

- Programmer द्वारा बनाया गया Function।

Function के लाभ

- Code Reuse
- Program छोटा और व्यवस्थित बनता है।
- Debugging आसान होती है।
- Maintenance आसान होती है।
- Program की Readability बढ़ती है।
- Modular Programming संभव होती है।

उदाहरण

```
#include<iostream>  
using namespace std;  
  
int add(int a,int b)  
{  
    return a+b;  
}  
  
int main()  
{  
    cout<<"Sum = "<<add(10,20);  
}
```

आउटपुट

Sum = 30

निष्कर्ष

Function प्रोग्राम को सरल, व्यवस्थित तथा पुनः उपयोग योग्य बनाता है। यह बड़े प्रोग्राम को छोटे भागों में विभाजित करके विकास और रखरखाव को आसान बनाता है।

Q8. What is a Loop in Programming? (10 Marks)

English Answer

Introduction

In programming, many tasks need to be performed repeatedly. Writing the same statements again and again increases program length. Loops solve this problem by executing a block of code multiple times.

Definition

A **loop** is a control structure that repeatedly executes a block of statements until a specified condition becomes false.

Types of Loops

1. for Loop

Used when the number of iterations is known.

```
for(int i=1;i<=5;i++)  
{  
    cout<<i;  
}
```

2. while Loop

Used when the condition is checked before execution.

```
while(i<=5)  
{  
    cout<<i;  
    i++;  
}
```

3. do-while Loop

Executes at least one time because the condition is checked after execution.

```
do  
{  
    cout<<i;  
    i++;  
}  
while(i<=5);
```

Advantages

- Reduces Code Repetition
- Saves Time
- Easy to Read
- Efficient Programming
- Improves Productivity

Disadvantages

- Infinite loop may occur.
- Wrong condition causes logical errors.
- Debugging may become difficult.

Applications

- Printing tables
- Array Traversal
- Pattern Printing
- Searching
- Mathematical Calculations

Conclusion

Loops simplify repetitive tasks and improve program efficiency. Choosing the correct loop makes programs shorter, faster, and easier to understand.

हिंदी उत्तर

परिचय

Programming में कई कार्य बार-बार करने पड़ते हैं। यदि प्रत्येक बार वही कोड लिखा जाए तो Program लंबा और जटिल हो जाता है। इस समस्या का समाधान Loop करता है।

परिभाषा

Loop एक Control Structure है जो किसी Condition के सत्य रहने तक Statements को बार-बार चलाता है।

Loop के प्रकार

1. for Loop

जब Iteration की संख्या पहले से ज्ञात हो।

2. while Loop

Condition पहले जाँची जाती है।

3. do-while Loop

कम-से-कम एक बार अवश्य चलता है।

Loop के लाभ

Q9. What is an Array in C and C++? (10 Marks)

English Answer

Introduction

Sometimes a program needs to store many values of the same type. Creating separate variables for each value is difficult. Arrays solve this problem by storing multiple values under one variable name.

Definition

An **array** is a collection of elements of the same data type stored in contiguous memory locations.

- Code Repetition कम होती है।

- समय की बचत।

- Program छोटा बनता है।

- Readability बढ़ती है।

- Efficiency बढ़ती है।

हानियाँ

- Infinite Loop हो सकता है।

- गलत Condition से Logical Error होती है।

- Debugging कठिन हो सकती है।

उपयोग

- Table Print करना

- Array Traversal

- Pattern Printing

- Searching

- Mathematical Problems

निष्कर्ष

Loop बार-बार होने वाले कार्यों को सरल बनाता है। यह Program को छोटा, तेज़ और अधिक प्रभावी बनाता है।

Each element is accessed using an index that starts from **0**.

Syntax

```
data_type array_name[size];
```

Example

```
int marks[5]={80,85,90,75,95};
```

Types of Arrays

1. One-Dimensional Array

2. Two-Dimensional Array
3. Multi-Dimensional Array

Advantages

- Stores multiple values using one variable.
- Easy data access using index.
- Reduces coding effort.
- Fast processing.
- Efficient memory utilization.

Disadvantages

- Fixed Size.
- Same data type only.
- Insertion and deletion are difficult.
- Wastage of memory if size is large.

Applications

- Student Marks
- Salary Records
- Matrix Operations
- Searching
- Sorting
- Data Processing

Example Program

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

int main()
{
    int a[5]={10,20,30,40,50};

    for(int i=0;i<5;i++)
        cout<<a[i]<<" ";

    return 0;
}
```

Output

10 20 30 40 50

Conclusion

Arrays are one of the most useful data structures in C and C++. They simplify the storage and processing of large amounts of similar data efficiently.

हिंदी उत्तर

परिचय

कई बार Program में समान प्रकार के अनेक मानों को संग्रहित करना होता है। प्रत्येक मान के लिए अलग Variable बनाना कठिन होता है। इस समस्या का समाधान Array करता है।

परिभाषा

Array समान Data Type के अनेक Elements का संग्रह है, जो Memory में लगातार (Contiguous) स्थानों पर संग्रहित होते हैं। प्रत्येक Element का Index 0 से प्रारम्भ होता है।

Syntax

```
data_type array_name[size];
```

उदाहरण

```
int marks[5]={80,85,90,75,95};
```

Array के प्रकार

1. One-Dimensional Array
2. Two-Dimensional Array
3. Multi-Dimensional Array

लाभ

- एक Variable में अनेक मान संग्रहित किए जा सकते हैं।
- Index द्वारा Data आसानी से प्राप्त होता है।
- Coding कम होती है।

- Processing तेज़ होती है।
- Memory का अच्छा उपयोग होता है।

हानियाँ

- Size निश्चित होता है।
- केवल एक ही Data Type के Elements रखे जा सकते हैं।
- Insert एवं Delete करना कठिन होता है।
- बड़ा Array होने पर Memory व्यर्थ हो सकती है।

उपयोग

- Student Marks
- Salary Records
- Matrix Operations
- Searching
- Sorting
- Data Processing

उदाहरण Program

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

int main()
{
    int a[5]={10,20,30,40,50};
    for(int i=0;i<5;i++)
        cout<<a[i]<<" ";

    return 0;
}
```

आउटपुट

10 20 30 40 50

निष्कर्ष

Array, C और C++ की सबसे महत्वपूर्ण Data Structures में से एक है। यह समान प्रकार के अनेक डेटा को एक ही Variable में व्यवस्थित रूप से संग्रहित करने की सुविधा देता है, जिससे Program अधिक सरल, तेज़ और प्रभावी बनता है।

Q10. What is Object-Oriented Programming (OOP) in C++? (10 Marks)

English Answer

Introduction

Object-Oriented Programming (OOP) is one of the most important programming paradigms used in C++. It organizes a program using **objects** and **classes** instead of only functions. OOP makes programs more secure, reusable, and easier to maintain. It is widely used for developing large software applications.

Definition

Object-Oriented Programming (OOP) is a programming methodology that uses **classes** and **objects** to design software. It combines data and functions into a single unit called an object and follows real-world concepts.

Basic Concepts of OOP

1. **Class** – A blueprint for creating objects.
2. **Object** – An instance of a class.
3. **Encapsulation** – Wrapping data and functions into one unit.
4. **Abstraction** – Hiding unnecessary details and showing only essential information.
5. **Inheritance** – Acquiring properties of one class into another.
6. **Polymorphism** – One interface with multiple forms.

Features of OOP

- Object-Based Programming
- Data Security

- Code Reusability
- Modularity
- Flexibility
- Easy Maintenance

Advantages of OOP

- Reduces code duplication.
- Makes programs secure.
- Easy debugging and maintenance.
- Improves software quality.
- Suitable for large projects.
- Promotes code reusability.

Disadvantages of OOP

- More complex than procedural programming.
- Requires more memory.
- Execution may be slightly slower.
- Not suitable for very small programs.

Applications of OOP

- Banking Systems
- Hospital Management
- School Management
- Game Development
- GUI Applications
- Real-Time Systems

Example Program

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

class Student
{
public:
    void display()
    {
        cout<<"Welcome to OOP";
    }
};

int main()
{
    Student s;
    s.display();
    return 0;
}
```

Output

Welcome to OOP

Conclusion

Object-Oriented Programming makes software more organized, reusable, secure, and easier to maintain. It is the foundation of modern software development.

हिंदी उत्तर परिचय

Object-Oriented Programming (OOP), C++ की सबसे महत्वपूर्ण प्रोग्रामिंग पद्धतियों में से एक है। इसमें प्रोग्राम को **Class** और **Object** के आधार पर बनाया जाता है। OOP बड़े सॉफ्टवेयर को सुरक्षित, व्यवस्थित तथा पुनः उपयोग योग्य बनाता है।

परिभाषा

Object-Oriented Programming (OOP) ऐसी प्रोग्रामिंग पद्धति है जिसमें **Class** और **Object** का उपयोग करके प्रोग्राम बनाया जाता है। इसमें Data और Functions को एक इकाई (Object) में रखा जाता है।

OOP के मुख्य सिद्धांत

1. Class
2. Object
3. Encapsulation
4. Abstraction
5. Inheritance
6. Polymorphism

OOP की विशेषताएँ

- Object आधारित Programming
- Data Security
- Code Reusability

- Modularity
- Flexibility
- आसान Maintenance

लाभ

- Code Reuse होता है।
- Program अधिक सुरक्षित होता है।
- Debugging आसान होती है।
- बड़े Software के लिए उपयुक्त।
- Maintenance आसान होती है।
- Software की गुणवत्ता बढ़ती है।

हानियाँ

- Procedural Programming की तुलना में अधिक जटिल।
- अधिक Memory की आवश्यकता।
- छोटे Programs के लिए उपयुक्त नहीं।
- Execution थोड़ा धीमा हो सकता है।

उपयोग

- Banking Software
- Hospital Management
- School Management
- Game Development
- GUI Applications

- Real-Time Systems

उदाहरण Program

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

class Student
{
public:
    void display()
    {
        cout<<"Welcome to OOP";
    }
};

int main()
{
    Student s;
    s.display();
}
```

आउटपुट

Welcome to OOP

निष्कर्ष

OOP आधुनिक सॉफ्टवेयर विकास की आधारशिला है। यह प्रोग्राम को सुरक्षित, व्यवस्थित, पुनः उपयोग योग्य तथा रखरखाव में आसान बनाता है।

Q11. What is a Class in C++? (10 Marks)

English Answer

Introduction

A class is the fundamental building block of Object-Oriented Programming. It defines the structure and behavior of objects. Classes help organize data and functions into a single unit.

Definition

A **class** is a user-defined data type that acts as a blueprint for creating objects. It contains **data members** (variables) and **member functions** (methods).

Syntax

```
class ClassName
{
private:
    // Data Members

public:
    // Member Functions
};
```

Components of a Class

1. Data Members
2. Member Functions
3. Access Specifiers
 - Private
 - Public
 - Protected

Features of Class

- User-defined data type
- Supports data hiding
- Promotes encapsulation
- Improves code organization
- Allows object creation

Advantages

- Data Security
- Code Reusability
- Easy Maintenance
- Better Organization
- Supports OOP

Example

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

class Student
{
public:
    int roll;
    void display()
    {
        cout<<"Roll Number = "<<roll;
    }
};

int main()
{
    Student s;
    s.roll=101;
    s.display();
}
```

Output

Roll Number = 101

Conclusion

A class is the blueprint of objects and is one of the most important concepts in C++. It helps create organized, secure, and reusable programs

हिंदी उत्तर (11)

परिचय

Class, Object-Oriented Programming का सबसे महत्वपूर्ण भाग है। यह Objects बनाने का आधार (Blueprint) होता है। इसमें Data तथा Functions को एक साथ रखा जाता है।

परिभाषा

Class एक **User-Defined Data Type** है, जो Object बनाने का Blueprint होता है। इसमें **Data Members** और **Member Functions** शामिल होते हैं।

Class के मुख्य भाग

1. Data Members
2. Member Functions
3. Access Specifiers
 - Private
 - Public
 - Protected

Class की विशेषताएँ

- User Defined Data Type
- Data Hiding
- Encapsulation
- बेहतर Code Organization
- Object Creation

लाभ

- Data सुरक्षित रहता है।
- Code Reuse होता है।
- Maintenance आसान होती है।
- Program व्यवस्थित रहता है।
- OOP को समर्थन मिलता है।

उदाहरण Program

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

```
class Student
{
public:
    int roll;

    void display()
    {
        cout<<"Roll Number = "<<roll;
```

```
};

int main()
{
    Student s;
    s.roll=101;
    s.display();
}
```

आउटपुट

Roll Number = 101

निष्कर्ष

Class, Object बनाने का आधार है। यह Data और Functions को एक साथ संगठित करके Program को अधिक सुरक्षित, व्यवस्थित तथा पुनः उपयोग योग्य बनाती है।

Q12. What is Inheritance in C++? (10 Marks)

English Answer

Introduction

Inheritance is one of the most important features of Object-Oriented Programming. It allows one class to acquire the properties and functions of another class. This helps reduce code duplication and increases software reusability.

Definition

Inheritance is the process by which one class (Derived Class) acquires the properties and methods of another class (Base Class).

Types of Inheritance

1. Single Inheritance
2. Multiple Inheritance
3. Multilevel Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

Advantages

- Code Reusability
- Easy Maintenance
- Faster Development
- Better Organization
- Extensibility
- Supports Real-World Relationships

Disadvantages

- Complex hierarchy may be difficult to understand.
- Multiple inheritance may cause ambiguity.
- Tight coupling between classes.

Example

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

class Animal
```

```
{
public:
    void eat()
    {
        cout<<"Animal Eats"<<endl;
    }
};

class Dog : public Animal
{
public:
    void bark()
    {
        cout<<"Dog Barks";
    }
};

int main()
{
    Dog d;
    d.eat();
    d.bark();
}
```

Output

Animal Eats
Dog Barks

Applications

- Banking Software
- Employee Management
- School Management
- Game Development
- Vehicle Management Systems

Conclusion

Inheritance is an essential feature of C++ that promotes code reuse, reduces duplication, and helps create organized software using parent-child relationships.

हिंदी उत्तर

परिचय

Inheritance, Object-Oriented Programming की एक महत्वपूर्ण विशेषता है। इसमें एक Class दूसरी Class के गुण (Properties) और Functions प्राप्त करती है। इससे Code Reusability बढ़ती है तथा Program अधिक व्यवस्थित बनता है।

परिभाषा

Inheritance वह प्रक्रिया है जिसमें **Derived Class**, **Base Class** के गुण और Functions प्राप्त करती है।

Inheritance के प्रकार

1. Single Inheritance
2. Multiple Inheritance
3. Multilevel Inheritance
4. Hierarchical Inheritance
5. Hybrid Inheritance

लाभ

- Code Reuse
- Maintenance आसान
- Development तेज़
- बेहतर Program Organization
- Extensibility
- Real-Life Relationships को दर्शाता है।

हानियाँ

- अधिक जटिल Hierarchy समझना कठिन हो सकता है।
- Multiple Inheritance में Ambiguity की समस्या हो सकती है।
- Classes के बीच अधिक Dependency हो सकती है।

उदाहरण Program

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

```
class Animal
{
public:
    void eat()
    {
        cout<<"Animal Eats"<<endl;
    }
};

class Dog : public Animal
{
public:
    void bark()
    {
        cout<<"Dog Barks";
    }
};

int main()
{
    Dog d;
    d.eat();
    d.bark();
}
```

आउटपुट

Animal Eats
Dog Barks

उपयोग

- Banking Software
- Employee Management
- School Management
- Game Development
- Vehicle Management System

निष्कर्ष

Inheritance, C++ की एक महत्वपूर्ण विशेषता है जो Code Reusability बढ़ाती है, Program को व्यवस्थित बनाती है तथा Parent-Child Relationship के माध्यम से बड़े Software के विकास को सरल बनाती है।

Q13. What is a Constructor in C++? (10 Marks)

English Answer

Introduction

A constructor is a special member function in C++ that is automatically called whenever an object of a class is created. It initializes the data members of an object and prepares it for use. Constructors simplify object initialization and improve program reliability.

Definition

A **constructor** is a special member function of a class that has the same name as the class and is automatically executed when an object is created. It has **no return type**, not even void.

Characteristics of Constructor

1. Same name as the class.

2. No return type.
3. Called automatically when an object is created.
4. Used to initialize object data.
5. Can be overloaded.

Types of Constructors

1. **Default Constructor** – Takes no arguments.
2. **Parameterized Constructor** – Accepts parameters.
3. **Copy Constructor** – Creates an object by copying another object.

Advantages

- Automatically initializes objects.
- Reduces coding effort.
- Improves code readability.
- Supports constructor overloading.
- Makes programs more reliable.

Example Program


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

class Student
{
public:
    Student()
    {
        cout<<"Constructor Called";
    }
};

int main()
{
    Student s;
    return 0;
}
```

Output

Constructor Called

Conclusion

Constructors are essential in C++ because they automatically initialize objects, making programs more organized, efficient, and easier to maintain.

हिंदी उत्तर

परिचय

Constructor, C++ का एक विशेष Member Function है, जो Object बनते ही अपने-आप (Automatically) कॉल होता है। इसका मुख्य कार्य Object के Data Members को प्रारम्भिक मान (Initialize) देना होता है।

परिभाषा

Constructor वह विशेष Member Function है जिसका नाम Class के नाम के समान होता है और जिसका कोई Return Type नहीं होता। Object के निर्माण के समय यह स्वतः Execute होता है।

Constructor की विशेषताएँ

1. Class के नाम के समान होता है।
2. कोई Return Type नहीं होता।
3. Object बनते ही स्वतः Call होता है।
4. Data Members को Initialize करता है।
5. Constructor Overloading का समर्थन करता है।

Constructor के प्रकार

1. Default Constructor
2. Parameterized Constructor
3. Copy Constructor

लाभ

- Object स्वतः Initialize होता है।
- Coding कम होती है।
- Program अधिक विश्वसनीय बनता है।
- Readability बढ़ती है।
- Constructor Overloading की सुविधा मिलती है।

उदाहरण Program

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

class Student
{
public:
    Student()
    {
        cout<<"Constructor Called";
    }
};

int main()
{
    Student s;
}
```

आउटपुट

Constructor Called

निष्कर्ष

Constructor, C++ का एक महत्वपूर्ण भाग है जो Object को स्वतः Initialize करता है तथा Program को अधिक सुरक्षित, व्यवस्थित और प्रभावी बनाता है।

Institute Of Computer Science & Technology (I.C.S.T)

कम्प्यूटर विज्ञान एवं प्रौद्योगिकी संस्थान (आई. सी. एस. टी.)

Q14. What is the use of printf() and scanf() in C? (10 Marks)
We Create **Qualified** IT Professionals

Introduction

Input and output are essential parts of every program. In the C programming language, the standard library provides two important functions—printf() and scanf()—to display output and accept input from the user.

Definition

- **printf()** is used to display formatted output on the screen.
- **scanf()** is used to accept formatted input from the keyboard.

Both functions are declared in the <stdio.h> header file.

Syntax

printf()

```
printf("format string", variables);
```

scanf()

```
scanf("format string",&variables);
```

Common Format Specifiers

Specifier Meaning

%d Integer

%f Float

%lf Double

%c Character

Rakesh Marg, Ghaziabad



- Easy input and output.
- Supports different data types.
- User-friendly interaction.
- Simple syntax.
- Widely used in C programming.

Example Program

```
www.icstinstitute.in  
#include<stdio.h>
```

```
int main()  
{  
    int age;  
  
    printf("Enter Age: ");  
    scanf("%d",&age);  
  
    printf("Age = %d",age);  
  
    return 0;
```

Output

Enter Age: 20
Age = 20
(An Autonomous Institution Registered Under the Society & Public Trust Act-1882 by Govt. of India, NCT of Delhi.)
A Division of THE INDIAN EDUCATIONAL TRUST (Reg. No.308)

Conclusion

printf() and scanf() are the most commonly used input/output functions in C. They help programmers interact with users efficiently and are fundamental to C programming.

UP Govt. & NCT of Delhi MSME
DOEACC (NIELIT) For CCC
ISO 21001:2018 Certified Institution

ADCA	DCA	DFA	BASIC	CCC	BCC
Web Designing	Tally GST	Python	JAVA .net	C/C++	SQL

For Any Query Contact Us On

9650571128 | 9910863033

किसी भी प्रोग्राम में Input और Output का महत्वपूर्ण स्थान होता है। C भाषा में `printf()` और `scanf()` सबसे अधिक उपयोग किए जाने वाले Standard Input/Output Functions हैं।

परिभाषा

- `printf()` का उपयोग स्क्रीन पर Output दिखाने के लिए किया जाता है।
- `scanf()` का उपयोग User से Input लेने के लिए किया जाता है।

दोनों Functions `<stdio.h>` Header File में परिभाषित हैं।

Syntax

printf()

```
printf("format string", variables);
```

scanf()

```
scanf("format string", &variables);
```

मुख्य Format Specifiers

Specifier उपयोग

%d Integer

%f Float

%lf Double

%c Character

लाभ

- Input एवं Output सरल बनते हैं।
- विभिन्न Data Types का समर्थन।
- User से आसानी से जानकारी प्राप्त होती है।
- Syntax सरल है।
- C Programming में सबसे अधिक उपयोग।

उदाहरण Program

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

```
int main()  
{
```

```
    int age;
```

```
    printf("Enter Age: ");
```

```
    scanf("%d", &age);
```

```
    printf("Age = %d", age);
```

```
    return 0;
```

```
}
```

आउटपुट

Enter Age: 20

Age = 20

निष्कर्ष

`printf()` और `scanf()` C भाषा के सबसे महत्वपूर्ण Input/Output Functions हैं। इनकी सहायता से Program और User के बीच आसानी से जानकारी का आदान-प्रदान किया जा सकता है।

Q15. What is the Difference Between = and == in C/C++? (10 Marks)

English Answer

Introduction

Operators are symbols that perform operations on variables and values. Among the most commonly used operators in C and C++, the **assignment operator (=)** and the **comparison operator (==)** are

often confused. Understanding the difference between them is essential for writing correct programs.

Definition

- **= (Assignment Operator):** Assigns a value to a variable.
- **== (Equality Operator):** Compares two values and returns **true** if they are equal; otherwise, it returns **false**.

Key Differences

Assignment Operator (=)	Comparison Operator (==)
Assigns value	Compares values
Changes variable value	Does not change value
Used in assignment statements	Used in conditions
Returns assigned value	Returns true or false

Example

```
int a = 10;
if (a == 10)
{
    cout<<"Equal";
}
```

Output

Equal

Advantages

- = stores values in variables.
- == helps in decision-making.
- Used in if, while, and for conditions.
- Essential for logical programming.

Common Mistake

Using = instead of == inside an if statement may produce incorrect results because = assigns a value instead of comparing it.

Conclusion

Both operators are important but serve different purposes. = is used for assigning values, whereas == is used for comparing values. Correct usage helps avoid logical errors.

हिंदी उत्तर

परिचय

Operators वे चिन्ह होते हैं जो Variables और Values पर विभिन्न प्रकार के कार्य करते हैं। C और C++ में = (Assignment Operator) तथा == (Comparison Operator) का सबसे अधिक उपयोग होता है। इन दोनों का कार्य अलग-अलग होता है।

परिभाषा

- **= (Assignment Operator):** किसी Variable को Value प्रदान करता है।
- **== (Comparison Operator):** दो Values की समानता की जाँच करता है। यदि दोनों समान हों तो **True**, अन्यथा **False** लौटाता है।

मुख्य अंतर

= (Assignment Operator) == (Comparison Operator)

Value Assign करता है Values की तुलना करता है

Variable का मान बदलता है Variable का मान नहीं बदलता

Assignment में उपयोग Condition में उपयोग

Assigned Value लौटाता है True या False लौटाता है

उदाहरण

```
int a = 10;  
if(a == 10)  
{  
    cout<<"Equal";  
}
```

आउटपुट

Equal

लाभ

- = द्वारा Variable में Value Store की जाती है।
- == द्वारा Decision Making की जाती है।
- if, while तथा for में व्यापक उपयोग।
- Logical Programming के लिए आवश्यक।

सामान्य त्रुटि

यदि if Statement में गलती से == की जगह = का उपयोग कर दिया जाए, तो Program गलत परिणाम दे सकता है क्योंकि वह तुलना करने के बजाय Value Assign कर देता है।

निष्कर्ष

= और == दोनों महत्वपूर्ण Operators हैं, लेकिन इनके कार्य अलग-अलग हैं। = का उपयोग Value Assign करने के लिए तथा == का उपयोग दो Values की तुलना करने के लिए किया जाता है। सही Operator का उपयोग Program को त्रुटिरहित और विश्वसनीय बनाता है।