Volume-8

RSSB

RAJASTHAN STAFF SELECTION BOARD

BASIC COMPUTER INSTRUCTOR

Key Features

- Covers Complete syllabus & all topics of Basic Computer Instructor Examination
- Thoroughly revised, fully solved & error free book
- Also Useful for IA, NIC, BCI & other Computer Exams
- Concise, concept oriented and topicwise presentation

National Board Helpline Number: + 91 809 444 1777

EA-Publications®
ENGINEERS ACADEMY PUBLICATIONS

W www.engineersacademy.org
E info@engineersacademy.org

Publisher and Distributor

Engineers Academy Publications

100-102, Ram Nagar, Bambala Puliya, Toll Tax,
Tonk Road, Pratap Nagar, Jaipur (Rajasthan)-302033
E-Mail : engineers.academy.india@gmail.com

All Rights Reserved :
This book or part there of cannot be translated or reproduced in any form (except for review or criticism) without the written permission from the Publishers.

First Edition : **2023**
Second Edition : **2026**

Without prior written permission of publisher and author, no person/publisher/institute should use full part of the text/design/question/material of the book. If any body/publisher/institute is found in default legal action will be taken accordingly.

Price : ₹ 700.00

Although every effort has been made to avoid mistakes and omissions, there may be possibility some mistakes been left inadvertently. This book is released with the understanding that neither author nor publisher will be responsible in any manner for mistakes/premissions in the book. Dispute, if any, shall be subject to Jaipur (Rajasthan) Jurisdiction only.

DIRECTOR'S *Message*

To reach heights one must start climbing and if the journey is difficult then perseverance is the key to success. As a teacher we have realized over past years that success in any competitive exam requires hard work and proper guidance. **Engineers Academy** with its unique teaching methodologies has always proved that we meet the expectations of thousands of students and parents to make their dreams come true. With changing patterns, we have adapted ourselves to deliver the best and ensure better results.

This book has been organized and executed with a lot of care, dedication and passion for lucidity. A conscious attempt has been made to simplify the concepts to facilitate better understanding of the subject.

Engineers Academy has many successful stories of students who secured All India Rank in ESE, GATE, PSUs and JEn. Now we invite you to become a part of Engineers Academy to explore and achieve ultimate goal of your life. We promise to provide you quality guidance with competitive environment which is far advanced and ahead than the reach of other institution.

We would feel satisfied if the book meets the needs of the students for whom it is meant.

Lastly, we are thankful to all the engineers, authors whose work has been the source of enlightenment, inspiration and guidance in presenting this book.

It is hoped that the book in its new form will enjoy its ever increasing popularity.

Regards

Dr. Pankaj Goyal

Preface

✐ This book has been written to meet the growing requirements of candidates appearing for Basic Computer Instructor and other competitive Examinations. Though every candidate has ability to succeed, but in today's competitive environment, in-depth knowledge, quality guidance, time management and good source of study is required to achieve goals.

This book covers the fundamental concepts of computer theory in a clear and structured manner. The topics in this book have been organized in systematic, chapterwise and topicwise manner, making them easy and interesting for even a beginner to understand. It is a very convenient book and must be studied by candidates preparing for competitive exams.

After studying this booklet students can feel encouraged and develop confidence to understand both theoretical and practical aspects of computers subjects. The concepts explained in a simple and effective way so that readers become well equipped to applied their knowledge in examinations as well as in practical solutions.

We hope this book will be proved an important tool to succeed in Basic Computer Instructor and other competitive Examinations.

Even though, enough efforts has been made for correcting errors and printing mistakes, due to human tendency there could be some minor typing mistakes in the book. If any such errors are found, they will be highly appreciated and in corporated in the next edition. Also, please provide your valuable suggestions at :engineers.academy.india@gmail.com

Wish you all the best. Have a nice reading.

Team of
Engineers Academy Publications

CONTENTS

S.No.	Paper	Page No.
1.	Programming Fundamentals	01–264
•	Clanguage	1–63
•	C++	63–100
•	Java	100–154
•	Net	154–172
•	Artificial Intelligence	172–187
•	Machine Learning	187–199
•	Python	199–245
•	Blockchain	246–262
•	Introduction of IDE	263–264

1
CHAPTER

Programming Language (प्रोग्रामिंग भाषा)

C भाषा क्या है? (What is C Language)

- C एक कंप्यूटर प्रोग्रामिंग भाषा है। इसे समझने से पहले यह जानना जरूरी है कि प्रोग्रामिंग भाषा क्या होती है।
- प्रोग्रामिंग भाषा वह माध्यम है जिसके द्वारा हम कंप्यूटर को निर्देश (instructions) देकर उससे कोई कार्य करवाते हैं।
- जिस प्रकार मनुष्य आपस में संवाद करने के लिए हिंदी, अंग्रेज़ी या अन्य भाषाओं का उपयोग करते हैं, उसी प्रकार कंप्यूटर से संवाद करने के लिए प्रोग्रामिंग भाषाओं का प्रयोग किया जाता है।
- हर भाषा की तरह प्रोग्रामिंग भाषाओं के भी कुछ निश्चित नियम होते हैं, जिन्हें Syntax (सिंटैक्स) कहा जाता है।
- इन सिंटैक्स नियमों का पालन करके हम कंप्यूटर को किसी विशेष कार्य के लिए निर्देश देते हैं और इसी प्रक्रिया से कंप्यूटर प्रोग्राम तैयार किए जाते हैं।
- C एक Structured (Procedural) और Middle-Level प्रोग्रामिंग भाषा है, जिसे सरलता और प्रभावशीलता के कारण काफी लोकप्रिय माना जाता है।
- इस भाषा का उपयोग मुख्य रूप से एप्लिकेशन सॉफ्टवेयर और सिस्टम सॉफ्टवेयर विकसित करने के लिए किया जाता है।
- C भाषा का विकास डेनिस रिची (Dennis Ritchie) द्वारा किया गया था।
- वर्ष 1978 में डेनिस रिची और ब्रायन कर्निघन (Brian Kernighan) ने C भाषा पर आधारित प्रसिद्ध पुस्तक "The C Programming Language" प्रकाशित की।
- C भाषा को कई बार कंप्यूटर भाषाओं की आधार भाषा (Mother Language) भी कहा जाता है, क्योंकि बाद में विकसित हुई कई भाषाएँ जैसे Java, PHP, C#, और C++ पर इसका प्रभाव देखा जाता है।

C Programming को Structured / Procedural Language क्यों कहा जाता है?

- C एक Structured या Procedural Programming Language है।
- इस प्रकार की भाषा में बड़े और जटिल प्रोग्राम को छोटे-छोटे भागों (modules) में विभाजित किया जाता है।
- इन छोटे भागों को Functions कहा जाता है।
- प्रत्येक function एक विशेष कार्य (specific task) को पूरा करता है।
- इस तरीके से प्रोग्राम को समझना, लिखना और संशोधित करना आसान हो जाता है।
- इसलिए C भाषा को Structured या Procedural Programming Language कहा जाता है।

C Programming को Middle Level Language क्यों कहा जाता है?

- C भाषा में Low Level Language के कुछ गुण होते हैं, जैसे कि यह memory को सीधे access (Direct Memory Access) कर सकती है।
- इसके साथ ही इसमें High Level Language के कई सुविधाजनक features भी होते हैं, जैसे functions, structured programming आदि।
- यानी C भाषा में Low Level और High Level दोनों प्रकार की विशेषताएँ मौजूद होती हैं।
- इसी कारण C को Middle Level Programming Language कहा जाता है।

C (सी) भाषा का इतिहास

- C प्रोग्रामिंग भाषा का विकास डेनिस रिची (Dennis Ritchie) द्वारा किया गया था।
- बाद में 1978 में ब्रायन कर्निघन (Brian Kernighan) और डेनिस रिची ने मिलकर C भाषा पर आधारित प्रसिद्ध पुस्तक "The C Programming Language" प्रकाशित की।
- समय के साथ C भाषा को बेहतर बनाने के लिए इसके कई नए संस्करण (versions) जारी किए गए।

प्रोग्रामिंग भाषाओं का विकास इतिहास | Development History of Programming Languages

भाषा / Language	वर्ष / Year	द्वारा विकसित / Developed By
Algol / Algol	1960	International Group
BCPL / BCPL	1966	Martin Richards
C / C	1969	Ken Thompson
C++ / C++	1972	Dennis Ritchie
ANSI C++ / ANSI C++ / ISO C++	1978	Brian Kernighan & Dennis Ritchie
C90	1989	ANSI Committee
C99	1990	ISO Committee
C11	1999	ISO Committee
C11	2011	Standardization Committee
C18	2018	Standardization Committee
C18	2018	Standardization Committee

C भाषा के प्रमुख संस्करण

1. **K & R C (Kernighan and Ritchie C)**
   - यह C भाषा का प्रारंभिक और मूल संस्करण माना जाता है।
   - इसे 1978 में प्रस्तुत किया गया था।
   - इस संस्करण में Standard Input/Output Library (I/O Functions) जैसी सुविधाएँ उपलब्ध थीं।
2. **ANSI C / ISO C**
   - इस संस्करण को American National Standards Institute (ANSI) द्वारा मानकीकृत किया गया।
   - बाद में International Organization for Standardization (ISO) ने भी इसे अपनाया।
   - यह मानक 1989–1990 के आसपास प्रकाशित किया गया था।
   - इसका उद्देश्य C भाषा के लिए एक समान और आधिकारिक मानक निर्धारित करना था।
3. **C99**
   - यह संस्करण 1999 में जारी किया गया।
   - इसमें कई नई सुविधाएँ जोड़ी गईं, जैसे:

- Inline Functions
- नए Data Types
- Long Integer (long int) जैसे विस्तृत डेटा प्रकार

4. C11
- यह संस्करण 2011 में प्रकाशित किया गया।
- इसमें C भाषा को और बेहतर बनाने के लिए कई नई विशेषताएँ जोड़ी गईं, जैसे:
  - नई Libraries
  - Type Generic Macros
  - Anonymous (अनाम) Structures
  - अन्य आधुनिक सुधार और सुविधाएँ।

C भाषा की प्रमुख विशेषताएँ (Features of C Language)

1. पोर्टेबिलिटी (Portability)
- C भाषा में लिखे गए प्रोग्राम अलग अलग-कंप्यूटर या प्लेटफॉर्म पर आसानी से चलाए जा सकते हैं।
- सामान्यतः प्रोग्राम को ज्यादा बदलाव किए बिना दूसरी मशीन पर भी उपयोग किया जा सकता है।
- विभिन्न सिस्टम पर Compiler और Preprocessor की सहायता से इसे चलाना संभव होता है।

2. शक्तिशाली प्रोग्रामिंग भाषा (Powerful Language)
- C भाषा को तेज और प्रभावी)efficient) प्रोग्रामिंग भाषा माना जाता है।
- इसमें Data Types, Functions और Control Statements जैसी सुविधाएँ उपलब्ध होती हैं, जिससे जटिल प्रोग्राम भी आसानी से बनाए जा सकते हैं।

3. सरल और समझने में आसान भाषा (Simple Language)
- C भाषा के अधिकांश commands अंग्रेजी शब्दों पर आधारित होते हैं।
- इसी कारण प्रोग्रामर के लिए कोड लिखना और समझना अपेक्षाकृत आसान हो जाता है।

4. Structured Oriented Language
- C एक Structured Programming Language है।
- इसमें बड़े प्रोग्राम को छोटे-छोटे भागों (functions / modules) में विभाजित किया जा सकता है।
- इससे प्रोग्राम की जटिलता कम होती है और कोड अधिक स्पष्ट व व्यवस्थित बनता है।

5. Compiler Based Language
- C भाषा में लिखे गए प्रोग्राम को सीधे रन नहीं किया जा सकता।
- पहले उसे Compiler की सहायता से Compile करना पड़ता है, उसके बाद ही प्रोग्राम को Execute किया जाता है।

6. Syntax Based Language
- C भाषा में Syntax (नियम और संरचना) का सख्ती से पालन करना आवश्यक होता है।
- यदि प्रोग्राम लिखते समय इन नियमों का पालन नहीं किया जाता, तो program में error आ जाता है।
- C, C++ और Java जैसी भाषाएँ Syntax आधारित भाषाओं के उदाहरण हैं।

7. Pointers का प्रभावी उपयोग
- C भाषा में Pointers का उपयोग किया जाता है।
- Pointer एक विशेष प्रकार का variable होता है जो किसी दूसरे variable का address (memory location) स्टोर करता है।
- इससे memory management और system programming अधिक प्रभावी बनती है।

8. Middle Level Language
- C भाषा में Low Level और High Level दोनों प्रकार की विशेषताएँ मौजूद होती हैं।
- इसलिए इसका उपयोग System Software और Application Software दोनों के विकास में किया जा सकता है।

9. Case Sensitive Language
- C भाषा Case Sensitive होती है।
- इसका मतलब छोटे और बड़े अक्षरों (lowercase और uppercase) को अलग-अलग माना जाता है।
- उदाहरण के लिए printf और PRINTF दोनों C भाषा में अलग माने जाते हैं।

C भाषा के अनुप्रयोग (Applications of C Language)

एप्लिकेशन सॉफ्टवेयर विकास
- C भाषा का उपयोग विभिन्न प्रकार के Application Software जैसे Database Programs, Spreadsheet Software आदि बनाने में किया जाता है।

Embedded Systems Development
- कई Embedded Systems और Embedded Software विकसित करने के लिए C भाषा का व्यापक रूप से उपयोग किया जाता है।

System Software निर्माण
- C भाषा का प्रयोग Operating System और अन्य System Software तैयार करने में भी किया जाता है।
- Graphics और Game Development
- कंप्यूटर और मोबाइल के लिए Graphics आधारित Applications और Games विकसित करने में भी C भाषा सहायक होती है।

Unix Kernel Development
- प्रसिद्ध Unix Operating System का Kernel मुख्य रूप से C भाषा में विकसित किया गया है।
- Network Devices और Device Drivers
- C भाषा का उपयोग Network Devices तथा Device Drivers बनाने में भी किया जाता है।

Compiler Development
- C भाषा की मदद से Compiler बनाए जाते हैं।
- Compiler का कार्य High Level Language में लिखे गए कोड को Machine Code या Low Level Code में बदलना होता है।
- आपके सिस्टम पर C भाषा की स्थापना (Installation of C on Your System)
- C भाषा के लिए कई प्रकार के Compiler उपलब्ध होते हैं।
- उपयोगकर्ता अपनी सुविधा के अनुसार किसी भी Compiler को डाउनलोड और इंस्टॉल कर सकता है।
- सामान्यतः Turbo C++ Compiler का उपयोग C और C++ दोनों प्रोग्राम लिखने और चलाने के लिए किया जाता है।

Turbo C++ इंस्टॉल करने की प्रक्रिया

1. Turbo C++ डाउनलोड करें
- सबसे पहले इंटरनेट से Turbo C++ Software डाउनलोड करें।
- यह सॉफ्टवेयर कई वेबसाइटों पर आसानी से उपलब्ध होता है।

2. C Drive में Directory बनाएं
- अपने कंप्यूटर की C Drive के अंदर एक नई turboc नाम की directory (फोल्डरबनाएं। (

3. Zip File Extract करें
- डाउनलोड की गई tc3.zip फ़ाइल को C:\turboc फोल्डर में extract (unzip) करें।

4. Installation शुरू करें

- Extract करने के बाद install.exe फ़ाइल पर डबल क्लिक करें।
- इससे Turbo C++ का installation process शुरू हो जाएगा।

5. Turbo C++ खोलें

- इंस्टॉलेशन पूरा होने के बाद C:\TC\BIN फोल्डर में जाएं।
- वहां मौजूद tc application (shortcut icon) पर क्लिक करके Turbo C++ शुरू किया जा सकता है।
- इसी में C program लिखे, compile और run किए जाते हैं।

Windows 7 / Windows 8 में संभावित समस्या

- Windows 7 या Windows 8 में Turbo C++ चलाते समय fullscreen mode supported नहीं होने का dialog box दिखाई दे सकता है।
- ऐसी स्थिति में Ignore बटन पर क्लिक करके प्रोग्राम को चलाया जा सकता है।

पहला C प्रोग्राम (First C Program)

- C भाषा में प्रोग्राम लिखने से पहले यह समझना जरूरी है कि C Console में प्रोग्राम को कैसे लिखा, compile और run किया जाता है।
- जब प्रोग्राम लिखा जाता है, तब उसे पहले compile किया जाता है और उसके बाद run करके परिणाम (Output) देखा जाता है।

First C Program बनाने से पहले ध्यान रखने योग्य बातें

Header File शामिल करना आवश्यक होता है।

- C प्रोग्राम में आवश्यक लाइब्रेरी का उपयोग करने के लिए header files को include किया जाता है।

Header File हमेशा # चिन्ह से शुरू होती है।

- उदाहरण के लिए #include निर्देश (directive) का उपयोग किया जाता है।

Program में main() function का होना अनिवार्य है।

- C प्रोग्राम का execution सामान्यतः main function से शुरू होता है।

Case Sensitivity का ध्यान रखना जरूरी है।

- C भाषा में uppercase और lowercase letters अलग-अलग माने जाते हैं, इसलिए कोड लिखते समय सही अक्षरों का प्रयोग करना चाहिए।

अधिकांश statements के अंत में Semicolon (;) लगाया जाता है।

- यदि statement के अंत में semicolon नहीं लगाया गया, तो program में error आ सकता है।

C Console खोलकर प्रोग्राम लिखा जाता है।

- इसके बाद console में C code टाइप करके program को compile और run किया जाता है।

पहला C प्रोग्राम (Example of First C Program)

C Program :

```
#include <stdio.h>

int main()
{
   printf("We are engineers academy");
   return 0;
}
```

Output : We are engineers academ

प्रोग्राम का संक्षिप्त विवरण

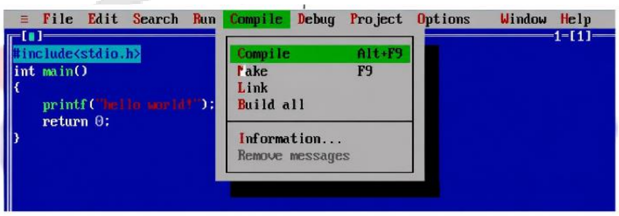
- #include <stdio.h>
  - यह एक header file है जिसमें Standard Input/Output Library Functions शामिल होते हैं।
  - printf() और scanf() जैसे functions इसी लाइब्रेरी में उपलब्ध होते हैं।
- int main()
  - यह C प्रोग्राम का मुख्य function (main function) होता है।
  - किसी भी C प्रोग्राम का execution सामान्यतः main() function से शुरू होता है।
- printf("Welcome to Technical Notes");
  - printf() का उपयोग screen पर message या Output दिखाने के लिए किया जाता है।
- return 0;
  - इसका अर्थ है कि प्रोग्राम सफलतापूर्वक समाप्त (successfully terminate) हो गया है।

C प्रोग्राम को Compile और Run कैसे करें

- C प्रोग्राम को Compile (संकलित) और Run (चलाने) के मुख्यतः दो तरीके होते हैं:
  - Menu के माध्यम से
  - Shortcut Keys के माध्यम से
- जब प्रोग्राम सफलतापूर्वक run होता है, तब उसका Output स्क्रीन पर दिखाई देता है।

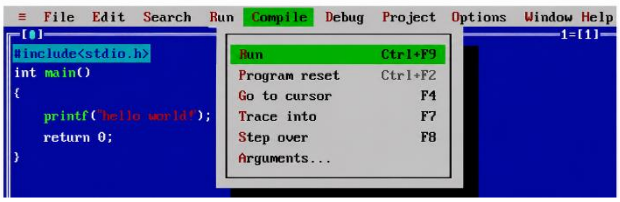
1. Menu के द्वारा C प्रोग्राम Compile और Run करना

- सबसे पहले C Console (जैसे Turbo C++) में अपना C प्रोग्राम लिखें।
- प्रोग्राम लिखने के बाद Compile Menu पर क्लिक करें।
- इसके बाद Compile Sub-Menu को चुनें, जिससे आपका C प्रोग्राम compile हो जाएगा।
- Compile होने के बाद Run Menu पर क्लिक करें।
- फिर Run Sub-Menu को चुनें, जिससे C प्रोग्राम execute (run) हो जाएगा और उसका Output स्क्रीन पर दिखाई देगा।



2. Shortcut Keys के द्वारा Program Run करना

- C प्रोग्राम को सीधे चलाने के लिए Ctrl + F9 कुंजियाँ दबाई जाती हैं।
- इसके बाद प्रोग्राम run हो जाएगा और Output user screen पर दिखाई देगा।



अतिरिक्त उपयोगी Shortcut Keys

Alt + F5

- इस कुंजी को दबाकर आप प्रोग्राम का Output screen पर कभी भी देख सकते हैं।

Esc Key

- Output देखने के बाद Turbo C++ Console पर वापस आने के लिए Esc कुंजी दबाई जाती है।

फ्लोचार्ट का परिचय (Introduction of Flowchart)

- फ्लोचार्ट किसी प्रोग्राम का चित्रात्मक (Graphical) प्रस्तुतीकरण होता है।
- इसके माध्यम से किसी प्रोग्राम की प्रक्रिया और लॉजिक को चित्रों तथा प्रतीकों (symbols) की मदद से समझाया जाता है।
- कई बार प्रोग्रामर सीधे C भाषा में प्रोग्राम लिख देते हैं, लेकिन केवल कोड देखकर प्रोग्राम की पूरी प्रक्रिया समझना कठिन हो सकता है।
- इसलिए फ्लोचार्ट की सहायता से प्रोग्राम के लॉजिक को ग्राफिकल रूप में प्रस्तुत किया जाता है, जिससे कोई भी व्यक्ति उसे आसानी से समझ सकता है।
- जब किसी कंपनी या संगठन में प्रोग्रामर के रूप में कार्य किया जाता है, तब अक्सर कोड लिखने से पहले फ्लोचार्ट तैयार किया जाता है।
- फ्लोचार्ट बनाने से प्रोजेक्ट की कार्यप्रणाली स्पष्ट हो जाती है, जिससे दूसरे लोग भी उस प्रोग्राम को आसानी से समझ सकते हैं और उस पर काम कर सकते हैं।
- फ्लोचार्ट की मदद से हमें प्रोग्राम की पूरी प्रक्रिया का बेहतर ज्ञान प्राप्त होता है।
- इससे कोडिंग शुरू करने से पहले संभावित त्रुटियों (errors) और परिणामों (Outputs) को समझना आसान हो जाता है।
- जब फ्लोचार्ट तैयार हो जाता है, तो उसके आधार पर कोड लिखना अधिक सरल और तेज हो जाता है, जिससे समय की बचत होती है।
- फ्लोचार्ट का एक महत्वपूर्ण लाभ यह है कि एक ही फ्लोचार्ट को देखकर अलग-अलग प्रोग्रामिंग भाषाओं में भी प्रोग्राम बनाया जा सकता है।
- फ्लोचार्ट को प्रोग्राम का ब्लूप्रिंट (Blueprint) भी कहा जाता है, क्योंकि यह प्रोग्राम की पूरी संरचना का प्रारूप प्रस्तुत करता है।
- फ्लोचार्ट बनाने में समय लगता है, लेकिन इससे प्रोग्राम को समझना और विकसित करना काफी आसान हो जाता है।
- सामान्यतः एल्गोरिथम (Algorithm) भी फ्लोचार्ट की प्रक्रिया के अनुसार ही तैयार किया जाता है।

फ्लोचार्ट के प्रतीक (Flowchart Symbols)

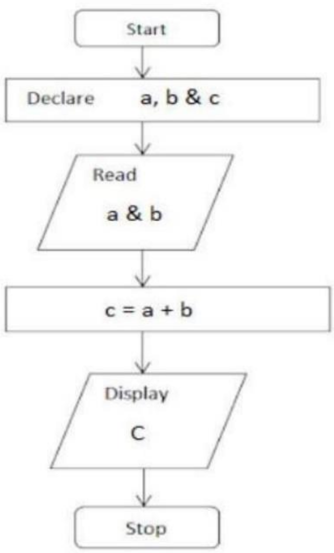
- किसी प्रोग्राम की प्रक्रिया को चित्रात्मक रूप (graphical form) में दर्शाने के लिए कुछ विशेष चिन्हों या आकृतियों का उपयोग किया जाता है।
- इन चिन्हों को Flowchart Symbols (फ्लोचार्ट प्रतीक) कहा जाता है।
- फ्लोचार्ट में प्रयुक्त प्रत्येक प्रतीक किसी विशेष क्रिया या प्रक्रिया को दर्शाता है।
- इन प्रतीकों की सहायता से प्रोग्राम के चरणों और उसके लॉजिक को आसानी से समझाया जा सकता है।
- जब कोई व्यक्ति फ्लोचार्ट को देखता है, तो वह प्रत्येक प्रतीक के अर्थ को समझकर प्रोग्राम की पूरी प्रक्रिया को सरलता से समझ सकता है।
- इस प्रकार फ्लोचार्ट प्रतीक प्रोग्राम के विभिन्न चरणों को व्यवस्थित और स्पष्ट रूप से प्रस्तुत करने में सहायता करते हैं।
- नीचे फ्लोचार्ट में उपयोग किए जाने वाले प्रमुख प्रतीकों का विवरण दिया जाता है।

फ्लोचार्ट प्रतिकों का विवरण

नाम	प्रतीक	व्याख्या
शुरू / अंत (Start/ End)		इस चिन्ह को अंडाकार चिन्ह या टर्मिनल चिन्ह भी कहा जाता है। इसका उपोग फ्लोचार्ट को **शुरू और सामात** करने के लिंए किया जाता है।
प्रसंस्करण (Processing)		इस प्रतीक को प्रसंस्करण प्रतीक कहा जाता है। इसका उपगिा किसी प्रसंस्करण (Processing) के लिए किया जाता है।
इनपुट / आउटपुट (input/ Output)		इस प्रतीक को **समांतर** चतुर्पुज प्रतीक कहा जाता है। इसका उपगिग इनपुट और आउटपुट को निरुपित करने के लिए किया जाता है।
फैसला (Decision)		इस प्रतीक को हीरा प्रतीक कहा जाता है। इसका उपगिग निर्पण (Decision) को निरुपित करने के लिए किया जाता है।
तीर	↓	इस प्रतीक को तीर का प्रतीक कहा जाता है। इस चिन्ह का प्रयेग कार्यक्रम के प्रवाह को दिखाने केलिए किया जाता है।

फ्लोचार्ट का उदाहरण (Example of Flowchart)

- मान लीजिए हमें ऐसा प्रोग्राम बनाना है जिसमें उपयोगकर्ता दो संख्याएँ (numbers) दर्ज करे और प्रोग्राम उन दोनों संख्याओं को जोड़कर परिणाम (result) प्रदर्शित करे।
- इस प्रकार के प्रोग्राम की प्रक्रिया को समझाने के लिए फ्लोचार्ट तैयार किया जाता है।
- इस उदाहरण में फ्लोचार्ट की शुरुआत अंडाकार (Oval) प्रतीक से की जाती है, जो Start को दर्शाता है।
- इसके बाद तीन variables – a, b और c को Rectangle (Process Symbol) की सहायता से घोषित किया जाता है।
- अगला चरण Input का होता है, जिसमें उपयोगकर्ता से a और b के मान (values) लिए जाते हैं।
- इसे फ्लोचार्ट में Parallelogram (Input/Output Symbol) द्वारा दर्शाया जाता है।
- इसके बाद दोनों संख्याओं को जोड़ने की प्रक्रिया की जाती है।
- इस प्रक्रिया में a + b का परिणाम c variable में संग्रहित (store) किया जाता है, जिसे Rectangle (Process Symbol) से दिखाया जाता है।
- अंत में परिणाम (sum) को प्रदर्शित किया जाता है, जिससे उपयोगकर्ता को दोनों संख्याओं का योग दिखाई देता है।



कैरेक्टर सेट (Character Set)

- वर्ण (Characters) वे मूल चिन्ह या अक्षर होते हैं जिनकी सहायता से किसी भी भाषा में शब्द बनाए जाते हैं।
- शब्दों के संयोजन से वाक्य बनते हैं, जो किसी विचार या अर्थ को व्यक्त करते हैं।

- कंप्यूटर भाषा भी एक प्रकार की भाषा है, जिसकी मदद से हम कंप्यूटर को अपनी आवश्यकताओं और निर्देशों के बारे में बताते हैं।
- अन्य भाषाओं की तरह कंप्यूटर भाषाओं में भी अपने नियम और व्याकरण (Grammar) होते हैं।
- किसी भी प्रोग्रामिंग भाषा को सीखने का पहला चरण उसके Character Set को समझना होता है।
- जिस प्रकार किसी नई भाषा को सीखते समय हमें पहले A, B, C, D जैसे अक्षर सिखाए जाते हैं, उसी तरह C भाषा सीखने से पहले उसके Character Set को जानना आवश्यक होता है।
- C भाषा में Character Set को मुख्य रूप से चार भागों में विभाजित किया जाता है।

1. अक्षर (Letters)

- अंग्रेजी के A से Z और a से z तक के अक्षर C भाषा में उपयोग किए जाते हैं।
- C भाषा Case Sensitive होती है, इसलिए A और a दोनों अलगअलग माने जाते हैं।-
- इन अक्षरों को मिलाकर Identifiers, Keywords और Strings आदि बनाए जाते हैं।

2. अंक (Digits)

- C भाषा में संख्यात्मक मान)Numeric Values) दर्शाने के लिए अंकों का प्रयोग किया जाता है।
- इसमें कुल 10 अंक)0,1,2,3,4,5,6,7,8,9) होते हैं।
- इन अंकों को समूह में भी उपयोग किया जा सकता है, जैसे:
  - 123
  - 345
  - 764 आदि।

3. विशेष वर्ण (Special Characters)

- C भाषा में कुछ विशेष चिन्ह)special symbols) भी उपयोग किए जाते हैं।
- इनका प्रयोग प्रोग्राम के विशेष भागों या विशेष कार्यों के लिए किया जाता है।
- इन विशेष वर्णों का उपयोग syntax और program structure को सही बनाने में महत्वपूर्ण होता है।

Symbol	Name
~	Tilde
!	Exclamation mark
#	Number sign
$	Dollar sign
%	Percent sign
:	Colon
'	Quotation mark
;	Semicolon
;	Quotation mark
<	Semicolon
<	Opening angle bracket
>	Closing angle bracket

चर (Variable)

- C भाषा में जब हमें किसी डेटा (data value) को प्रोग्राम में उपयोग करना होता है, तो उसे मेमोरी में स्टोर किया जाता है।
- उस मेमोरी स्थान (memory location) को एक नाम दिया जाता है, जिससे उसे आसानी से उपयोग या एक्सेस किया जा सके।
- मेमोरी के किसी स्थान को दिया गया नाम ही Variable कहलाता है।
- दूसरे शब्दों में, Variable मेमोरी लोकेशन का नाम होता है।
- स्थिरांक (Constant) के विपरीत, Variable का मान बदला जा सकता है।
- प्रोग्राम के execution के दौरान Variable की value को आवश्यकता अनुसार बदला जा सकता है।
- प्रोग्रामर आमतौर पर अर्थपूर्ण (meaningful) variable names का उपयोग करता है, जैसे:
  - average
  - height
  - age
  - total आदि।

वेरिएबल का Data Type

- C भाषा में प्रत्येक Variable को एक Data Type देना आवश्यक होता है।
- Data Type यह निर्धारित करता है कि Variable किस प्रकार का डेटा संग्रहित करेगा।

मुख्य Data Types इस प्रकार हैं:

char

- किसी एक character (अक्षर) को store करने के लिए उपयोग किया जाता है।

int

- पूर्णांक (integer values) को store करने के लिए प्रयुक्त होता है।

float

- दशमलव (decimal) या floating point values को store करने के लिए उपयोग किया जाता है।

double

- बड़े या अधिक सटीक decimal values को store करने के लिए प्रयोग किया जाता है।

void

- इसका उपयोग तब किया जाता है जब कोई value return या store नहीं करनी होती।

Variable Name रखने के नियम

- Variable का नाम किसी अंक (digit) से शुरू नहीं होना चाहिए।
- Variable name में अक्षर (letters), अंक (digits) और underscore _ का उपयोग किया जा सकता है।
- Variable name में space (खाली स्थान) की अनुमति नहीं होती।
- Keywords को Variable name के रूप में उपयोग नहीं किया जा सकता।
- C भाषा Case Sensitive होती है, इसलिए Uppercase और Lowercase नाम अलग माने जाते हैं।
- उदाहरण: total और TOTAL दोनों अलग हैं।
- सामान्यतः सुझाव दिया जाता है कि variable names को lowercase में रखा जाए।

Variable Declaration, Definition और Initialization

- किसी भी Variable का उपयोग करने से पहले उसे program में declare करना आवश्यक होता है।
- Variable की Declaration से निम्न जानकारी मिलती है:
  - Compiler को बताया जाता है कि Variable का नाम क्या है।
  - यह भी बताया जाता है कि Variable किस प्रकार का डेटा store करेगा)Data Type)।

- जब तक Variable को define नहीं किया जाता, तब तक compiler को उसके लिए memory allocate करने की आवश्यकता नहीं होती।
- Declaration का अर्थ है कि compiler को यह जानकारी देना कि program में एक variable मौजूद है और उसका data type क्या है।
- यदि किसी variable को main() function के बाहर घोषित किया जाता है, तो उसे extern keyword की सहायता से declare किया जा सकता है।

Expression (व्यंजक)

- Expression वह संयोजन होता है जिसमें Variables (चर), Constants (स्थिरांक), Operators (ऑपरेटर) और Function Calls शामिल होते हैं।
- C भाषा में Expression का उपयोग किसी मान की गणना (calculation) करने या परिणाम प्राप्त करने के लिए किया जाता है।
- Expression विभिन्न प्रकार के हो सकते हैं, जैसे:
  - Arithmetic Expression (अंकगणितीय व्यंजक)
  - Relational Expression (संबंधपरक व्यंजक)
  - Logical Expression (तार्किक व्यंजक)
  - Function Call Expression

  उदाहरण :
  - int z = x + y → Arithmetic Expression
  - a > b → Relational Expression
  - a == b → Logical Expression
  - func(a, b) → Function Call Expression

Constant Expression (अचर व्यंजक)

- वह Expression जो केवल स्थिर मानों (constants) से मिलकर बना हो, उसे Constant Expression कहा जाता है।
- उदाहरण:
  - 121 + 17 - 110
  - यह एक Constant Expression है क्योंकि इसमें सभी मान स्थिर)constant) हैं।
- लेकिन यदि Expression में कोई variable शामिल हो, तो वह Constant Expression नहीं माना जाता।
- **उदाहरण**:
  - 180 + 2 - j
  - यहाँ j एक variable है, इसलिए यह Constant Expression नहीं है।

C में कीवर्ड (Keywords in C)

- जिस प्रकार हमारी सामान्य भाषा (जैसे हिंदी) में कुछ शब्दों का पहले से निश्चित अर्थ होता है, जैसे – खेलो, पढ़ो, सुनाओ, इसलिए इन शब्दों का उपयोग किसी अन्य अर्थ में नहीं किया जाता।उसी प्रकार प्रोग्रामिंग भाषा में भी कुछ विशेष शब्द होते हैं, जिन्हें Keywords (कीवर्ड) कहा जाता है।
- Keywords आरक्षित शब्द (Reserved Words) होते हैं, जिनका C प्रोग्रामिंग भाषा में पहले से निर्धारित अर्थ और कार्य होता है।
- क्योंकि इन शब्दों का अर्थ पहले से तय होता है, इसलिए इन्हें किसी अन्य उद्देश्य जैसे variable name, function name आदि के रूप में उपयोग नहीं किया जा सकता।
- C प्रोग्रामिंग भाषा में कुल 32 Keywords होते हैं।
- C भाषा में सभी Keywords हमेशा lowercase (छोटे अक्षरों) में लिखे जाते हैं।
- नीचे C भाषा के 32 Keywords की सूची दी जाती है।

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	Volatile
const	float	short	Unsigned

C में Identifiers का अर्थ (Meaning of Identifiers in C)

- जब हम कोई प्रोग्राम विकसित (develop) करते हैं, तब हमें कंप्यूटर की मेमोरी में विभिन्न प्रकार के डेटा को इनपुट करना और उस पर प्रोसेसिंग करना पड़ता है।
- किसी भी प्रकार की प्रक्रिया करने से पहले डेटा को कंप्यूटर की मेमोरी में स्टोर करना आवश्यक होता है।
- यदि डेटा मेमोरी में स्टोर नहीं किया गया है, तो उस पर कोई भी प्रक्रिया (processing) नहीं की जा सकती।
- कंप्यूटर की मेमोरी के प्रत्येक स्थान (memory location) का एक विशिष्ट पता (unique address) होता है।
- जब हम कंप्यूटर में कोई डेटा इनपुट करते हैं, तो वह डेटा मेमोरी के किसी स्थान पर जाकर संग्रहित (store) हो जाता है।
- सामान्य रूप से हम यह नहीं जान सकते कि हमारा डेटा किस मेमोरी लोकेशन पर स्टोर हुआ है, क्योंकि मेमोरी आवंटन का कार्य ऑपरेटिंग सिस्टम (Operating System) द्वारा किया जाता है।
- जब compiler किसी डेटा को स्टोर करने के लिए मेमोरी सुरक्षित करता है, तब हम उस मेमोरी स्थान को एक नाम देते हैं।
- इसी नाम की सहायता से हम मेमोरी में संग्रहित डेटा को पहचान और उपयोग कर सकते हैं।
- किसी डेटा की मेमोरी लोकेशन को दिया गया नाम ही Identifier कहलाता है।
- Identifier के नाम कुछ नियमों (rules) के अनुसार बनाए जाते हैं, क्योंकि C compiler इन्हीं नियमों के आधार पर प्रोग्राम को समझता और प्रोसेस करता है।

Identifier Naming Rules (Identifier Naming Convention)

- Identifier के नाम में Uppercase और Lowercase दोनों प्रकार के अक्षरों का उपयोग किया जा सकता है।
- Identifier के नाम में Underscore _ का भी उपयोग किया जा सकता है।
- यदि Identifier के नाम में digits (अंक) का उपयोग किया जाता है, तो उससे पहले कम से कम एक अक्षर या underscore होना चाहिए।
- Identifier के नाम में विशेष चिन्ह जैसे – dot (.), comma (,), space आदि का उपयोग नहीं किया जा सकता।
- किसी भी Reserved Word (Keyword) या Built-in Function के नाम को Identifier के रूप में उपयोग नहीं किया जा सकता।
- Identifier का नाम किसी अंक (digit) से शुरू नहीं होना चाहिए।
- %, @, - जैसे symbols का उपयोग Identifier में नहीं किया जाता।
- C भाषा Case Sensitive होती है, इसलिए num और Num दो अलग-अलग identifiers माने जाते हैं।
- Identifier के नाम में operators का उपयोग नहीं किया जा सकता।

- Identifier हमेशा किसी अक्षर (letter) या underscore _ से शुरू होना चाहिए।

Identifier के उदाहरण

- _ram → सही (Right)
- ram-123 → गलत (Wrong)
- ram_45 → सही (Right)
- 4ram → गलत (Wrong)

Data Types (डेटा टाइप)

- जब हम C language में प्रोग्राम लिखते हैं, तब हमें विभिन्न प्रकार के variables (चर) बनाने पड़ते हैं।
- इन variables के प्रकार को ही Data Type कहा जाता है।
- किसी भी प्रोग्रामिंग भाषा जैसे C या C++ में प्रोग्राम बनाने के लिए Data Type घोषित करना आवश्यक होता है।
- यदि हम C भाषा में कोई प्रोग्राम बनाना चाहते हैं, तो सबसे पहले variables बनाकर उनके Data Types निर्धारित करने पड़ते हैं।

Data Type Modifiers (डेटा टाइप संशोधक)

- जैसा कि नाम से स्पष्ट है, Data Type Modifiers का उपयोग मूल (basic) data types की क्षमता या आकार को बदलने के लिए किया जाता है।
- Modifiers वास्तव में C भाषा के keywords होते हैं, जो किसी basic data type के अर्थ या सीमा को बदल देते हैं।
- इनका उपयोग यह बताने के लिए किया जाता है कि किसी variable के लिए कितनी memory allocate की जाएगी।
- Modifiers को सामान्यतः basic data types के पहले (prefix) लगाकर variable के लिए आवंटित memory को संशोधित किया जाता है।

C में Data Type Modifiers के प्रकार

- C प्रोग्रामिंग भाषा में मुख्य रूप से 5 प्रकार के Data Type Modifiers होते हैं:
  1. long
  2. short
  3. signed
  4. unsigned
  5. long long

C भाषा के मुख्य Data Types

C भाषा में मुख्य रूप से निम्न Data Types उपयोग किए जाते हैं:

int (Integer)

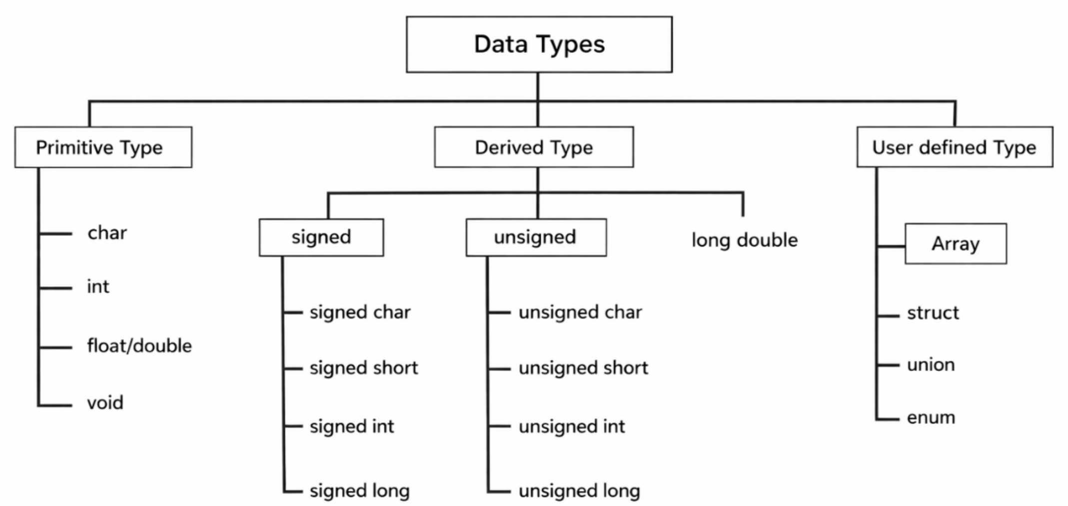
- पूर्णांक (whole numbers) को store करने के लिए उपयोग किया जाता है।

float

- दशमलव (decimal) या floating point numbers को store करने के लिए उपयोग किया जाता है।

char (Character)

- किसी एक अक्षर या character को store करने के लिए उपयोग किया जाता है।



C भाषा में Data Types के प्रकार

C language में मुख्य रूप से तीन प्रकार के data types होते हैं –

1. पूर्व-परिभाषित (Predefined / Primitive) Data Types
2. Derived Data Types
3. उपयोगकर्ता-परिभाषित (User Defined) Data Types

1. आदिम डेटा प्रकार (Primitive Data Types)

- Primitive Data Types वे डेटा प्रकार होते हैं जो पहले से परिभाषित)Built-in / Predefined) होते हैं।
- इनका उपयोग प्रोग्रामर द्वारा सीधे variables घोषित करने के लिए किया जाता है।
- C भाषा में कई predefined data types उपलब्ध होते हैं, जिनमें से एक प्रमुख प्रकार Integer Data Type है।

Integer Data Type (पूर्णांक प्रकार)

- Integer प्रकार का variable बनाने के लिए int keyword का उपयोग किया जाता है।
- Integer data type का उपयोग पूर्ण संख्याएँ (whole numbers / numeric values) स्टोर करने के लिए किया जाता है।
- int data type से बना variable 2 byte, 4 byte या 8 byte तक डेटा स्टोर कर सकता है, लेकिन यह compiler और system architecture पर निर्भर करता है।
- कुछ कंपाइलर में int variable 2 bytes memory लेता है, जबकि कुछ में 4 bytes memory का उपयोग होता है।

Integer Data Type की Value Range

int (2 byte)

- यह -32,768 से 32,767 तक के मान (values) स्टोर कर सकता है।

int (4 byte)

- यह -2,147,483,648 से 2,147,483,647 तक के मान स्टोर कर सकता है।

Data Types	Memory Size	Range
char	1 byte	−128 to 127
signed char	1 byte	−128 to 127
unsigned char	1 byte	0 to 255
short	2 bytes	−32,768 to 32,767
signed short	2 bytes	−32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
int	2 bytes	−32,768 to 32,767
signed int	2 bytes	−32,768 to 32,767
unsigned int	2 bytes	0 to 65,535
short int	2 bytes	−32,768 to 32,767
signed short int	2 bytes	0 to 65,535
long int	4 bytes	−2,147,483,648 to 2,147,483,647
signed long int	4 bytes	−2,147,483,648 to 2,147,483,647
unsigned long int	4 bytes	0 to 4,294,967,295
float	4 bytes	1.2E-38 to 3.4E+38
double	8 bytes	2.3E-308 to 1.7E+308
long double	10 bytes	3.4E-4932 to 1.1E+4932
long double	10 bytes	3.4E-4932 to 1.1E+4932

Long Integer

- यदि हमें int की सीमा से अधिक बड़ी संख्याएँ स्टोर करनी हों, तो हम long int keyword का उपयोग कर सकते हैं।
- long int से बना variable सामान्य int variable की तुलना में अधिक डेटा स्टोर करने की क्षमता रखता है।

Format Specifier

- %d format specifier का उपयोग int data type के variable का मान print करने के लिए किया जाता है।

सिंटैक्स -:

```
 int r= 5;
long int t ;
```

उदाहरण

```
#include  < stdio.h > int main()
{
Int p = 5;
 long int q;
printf ( " enter a Number \ n " ) ;
scanf ("% d",& q);
 printf(" value p = % d , q = %d ", p, q); return 0;
 }
```

Output:

एक नंबर दर्ज करें

10

p का मान = 5 , q= 10

Character Data Type (चरित्र डेटा प्रकार)

- Character Data Type का उपयोग अक्षर (characters) को स्टोर करने के लिए किया जाता है।
- C भाषा में character type का variable बनाने के लिए char keyword का प्रयोग किया जाता है।
- char से बना variable सामान्यतः केवल एक single character को store करता है।
- Character data type का size 1 byte होता है।

%c format specifier का उपयोग char data type के variable का मान print करने के लिए किया जाता है।

Syntax

```
char ch1 = 'A';
char ch2 = 'a';
```

उदाहरण

```
#include < stdio.h >
int main()
{
char ch1 = 'A';
char ch2;
printf ( " enter a character \ n " ) ;
scanf("% c",&ch 2);
printf ("value" of ch1 = %c , ch2= %c ", ch1 , ch2 ) ;
return 0;
}
```

आउटपुट -:

एक character दर्ज करें

g

value of ch1 = A , ch2 = g

फ़्लोटिंग पॉइंट डेटा प्रकार (Floating Point Data Types)

- Floating Point Data Types का उपयोग दशमलव (decimal) या fractional values को स्टोर करने के लिए किया जाता है।
- C भाषा में मुख्य रूप से दो प्रकार के Floating Point Data Types होते हैं:
  - float
  - double
- इन डेटा प्रकारों का उपयोग उन संख्याओं के लिए किया जाता है जिनमें दशमलव बिंदु (decimal point) होता है।

Float Data Type

- float keyword का उपयोग floating point type का variable बनाने के लिए किया जाता है।
- float data type से बना variable सामान्यतः 4 bytes memory लेता है।
- %f format specifier का उपयोग float data type के variable का मान print करने के लिए किया जाता है।

फ्लोट डेटा प्रकार का उदाहरण

```
#include <stdio.h>
int main()
{float q= 5.50;
float w;
printf ( " enter a decimal Number w = \ n " ) ;scanf
("%f",&w);
printf (" q और w का योग %f है ", q + w); return 0;
}
```

आउटपुट -:

एक दशमलव संख्या दर्ज करें w

= 10.06

q और w का योग 15.560000

Double Data Type

- Double Data Type का उपयोग भी दशमलव (decimal) मानों को स्टोर करने के लिए किया जाता है, ठीक उसी प्रकार जैसे float data type का उपयोग किया जाता है।

- C भाषा में double प्रकार का variable बनाने के लिए double keyword का उपयोग किया जाता है।
- double data type से बना variable सामान्यतः 8 bytes memory लेता है।
- double data type का उपयोग तब किया जाता है जब अधिक सटीक (precision) या बड़े decimal values को store करना हो।
- %lf format specifier का उपयोग double data type से बने variable का मान print करने के लिए किया जाता है।

डबल डेटा प्रकार का उदाहरण

```
 #include <stdio.h>
int main()
{
फ्लोट j = 679999999.454;
डबल k = 679999999.454;
printf("फ्लोट j = %f और डबल k = % lf", x, y); return 0;
}
```

उत्पादन(Output**)**

फ्लोट j= 68000000000000 और डबल k = 679999999.454000

Note :- ऊपर हमने जो डाटा टाइप्स सीखे हैं उन्हें बेसिक डेटा टाइप्स या प्री- डिफाइन्ड डेटा टाइप्स कहते हैं।

व्युत्पन्न डेटा प्रकार (Derived Data Types)

- वे डेटा प्रकार जो Primitive या Built-in Data Types से बनाए जाते हैं, उन्हें Derived Data Types कहा जाता है।
- ये डेटा प्रकार मूल डेटा प्रकारों के आधार पर अधिक जटिल डेटा संरचनाएँ बनाने के लिए उपयोग किए जाते हैं।
- C/C++ में Derived Data Types के मुख्य प्रकार इस प्रकार हैं:
    - Function (फंक्शन)
    - Array (ऐरे)
    - Pointer (पॉइंटर)
    - Reference (संदर्भ)

उपयोगकर्ता-परिभाषित डेटा प्रकार (User Defined Data Types)

- ऐसे डेटा प्रकार जिन्हें प्रोग्रामर या उपयोगकर्ता स्वयं परिभाषित करता है, उन्हें User Defined Data Types कहा जाता है।
- इनका उपयोग प्रोग्राम में नए और जटिल डेटा स्ट्रक्चर बनाने के लिए किया जाता है।
- C++ भाषा में कुछ प्रमुख User Defined Data Types इस प्रकार हैं:
    - Class (वर्ग)
    - Structure (संरचना)
    - Union (संघ)
    - Enumeration (गणना)
- इसके अतिरिक्त Type Defined Data Type (typedef) का उपयोग भी नए नाम से डेटा प्रकार परिभाषित करने के लिए किया जाता है।

Void Data Type (शून्य डेटा प्रकार)

- void डेटा प्रकार का अर्थ होता है "कोई मान नहीं" या "कोई प्रकार नहीं"।
- void data type का उपयोग सामान्य variables के साथ नहीं किया जा सकता।
- इसका उपयोग अधिकतर Function Return Type के रूप में किया जाता है।
- यदि किसी function का return type void होता है, तो इसका अर्थ है कि वह कोई भी value return नहीं करेगा।
- void data type का उपयोग functions और pointers के साथ अधिक स्पष्ट रूप से समझ में आता है।

ऑपरेटर्स (Operators)

- C भाषा में Operators का बहुत महत्वपूर्ण स्थान होता है।
- जब भी हम Variables या Values के साथ किसी प्रकार की गणना या प्रक्रिया करते हैं, तब Operators का उपयोग किया जाता है।
- C प्रोग्रामिंग में शर्तें (conditions) बनाने और विभिन्न प्रकार के ऑपरेशन करने के लिए Operators आवश्यक होते हैं।
- बिना Operators के किसी भी प्रोग्राम में गणना या लॉजिक लागू करना संभव नहीं होता।

C भाषा में Operators के प्रकार

C भाषा में निम्न प्रकार के Operators पाए जाते हैं:

- Arithmetic Operators (अंकगणितीय ऑपरेटर)
- Relational Operators (संबंधपरक ऑपरेटर)
- Logical Operators (तार्किक ऑपरेटर)
- Bitwise Operators (बिटवाइज ऑपरेटर)
- Assignment Operators (असाइनमेंट ऑपरेटर)
- Increment Operators (इंक्रीमेंट ऑपरेटर)
- Decrement Operators (डिक्रीमेंट ऑपरेटर)
- Conditional Operator (कंडीशनल ऑपरेटर)
- Special Operators (विशेष ऑपरेटर)
- Access Operators (एक्सेस ऑपरेटर)

1. Arithmetic Operators (अंकगणितीय ऑपरेटर)

- Arithmetic Operators का उपयोग गणितीय क्रियाएँ करने के लिए किया जाता है, जैसे:
    - जोड़ (Addition)
    - घटाव (Subtraction)
    - गुणा (Multiplication)
    - भाग (Division)

  Arithmetic Operators के उदाहरण:

  + , - , * , / , % , ++ , --

Arithmetic Operators के प्रकार

1. Unary Operators (यूनरी ऑपरेटर)
   - वे ऑपरेटर जो केवल एक operand पर कार्य करते हैं, उन्हें Unary Operators कहा जाता है।

2. Binary Operators (बाइनरी ऑपरेटर)
   - वे ऑपरेटर जो दो operands पर कार्य करते हैं, उन्हें Binary Operators कहा जाता है।

मुख्य Arithmetic Operators

Addition (+)

- दो variables या numbers को जोड़ने के लिए उपयोग किया जाता है।

Subtraction (-)

- एक value में से दूसरी value को घटाने के लिए उपयोग किया जाता है।

Multiplication (*)

- दो values का गुणा करने के लिए प्रयोग किया जाता है।

Division (/)

- एक value को दूसरी value से भाग देने के लिए उपयोग किया जाता है।

Modulus (%)

- यह ऑपरेटर division के बाद शेष (remainder) प्राप्त करने के लिए उपयोग किया जाता है।

2. Relational Operators (संबंधपरक ऑपरेटर)

- Relational Operators का उपयोग दो values के बीच तुलना)comparison) करने के लिए किया जाता है।
- इनका परिणाम हमेशा True (1) या False (0) के रूप में प्राप्त होता है।
- Relational Operators के उदाहरण:

== , != , < , > , <= , >=

Relational Operators का विवरण

- Equal To (==)
  - यह जाँचता है कि दोनों values समान हैं या नहीं।
- Not Equal To (!=)
  - यह जाँचता है कि दोनों values असमान हैं या नहीं।
- Less Than (<)
  - यह जाँचता है कि बाएँ operand का मान दाएँ operand से कम है या नहीं।
- Greater Than (>)
  - यह जाँचता है कि बाएँ operand का मान दाएँ operand से अधिक है या नहीं।
- Less Than or Equal To (<=)
  - यह जाँचता है कि बाएँ operand कम या बराबर है या नहीं।
- Greater Than or Equal To (>=)
  - यह जाँचता है कि बाएँ operand अधिक या बराबर है या नहीं।

3. Logical Operators (लॉजिकल ऑपरेटर)

- Logical Operators का उपयोग दो या दो से अधिक conditions को जोड़ने के लिए किया जाता है।
- इनका परिणाम हमेशा Boolean Value में होता है:
  - True = 1
  - False = 0

Logical Operators

- AND (&&)
  - जब दोनों conditions true होती हैं तभी result true होता है।
- OR (||)
  - यदि कम से कम एक condition true है तो result true होता है।
- NOT (!)
  - यह operator condition के result को उल्टा (reverse) कर देता है।

4. Bitwise Operators (बिटवाइज ऑपरेटर)

- Bitwise Operators का उपयोग बिट स्तर)bit level) पर ऑपरेशन करने के लिए किया जाता है।
- इसमें values को पहले binary form में बदलकर फिर ऑपरेशन किया जाता है।

Bitwise Operators

- Bitwise AND (&)
  - दो binary numbers के bits पर AND operation करता है।
- Bitwise OR (|)
  - binary bits पर OR operation करता है।
- Bitwise NOT (~)
  - यह bits को reverse कर देता है)0 → 1 और 1 → 0)।
- Bitwise XOR (^)
  - यदि bits अलग हों तो result 1 और समान हों तो 0 देता है।
- Left Shift (<<)
  - यह bits को बाईं ओर shift करता है।
- Right Shift (>>)
  - यह bits को दाईं ओर shift करता है।

5. Assignment Operators (असाइनमेंट ऑपरेटर)

- Assignment Operators का उपयोग variables को value assign करने के लिए किया जाता है।
- इसमें left side operand variable होता है और right side operand value होती है।

Assignment Operators

- =
  - Right side value को left side variable में store करता है।
- +=
  - Left variable में right value जोड़कर result store करता है।
- -=
  - Left variable से right value घटाकर result store करता है।
- *=
  - Left और right value को multiply करके result store करता है।
- /=
  - Left value को right value से divide करके result store करता है।
- %=
  - Division के बाद remainder को variable में store करता है।

इंक्रीमेंट / डिक्रीमेंट ऑपरेटर्स

- C भाषा में किसी भी वेरिएबल की वैल्यू को बढ़ाने या घटाने के लिए Increment और Decrement Operators का प्रयोग किया जाता है।
- इन ऑपरेटरों की सहायता से वेरिएबल के मान को एक इकाई (1) से बढ़ाया या कम किया जा सकता है।

Increment (वृद्धि) ++

- Increment Operator का उपयोग किसी expression में variable की value को बढ़ाने के लिए किया जाता है।
- Pre-Increment में पहले variable की value increase (increment) की जाती है और उसके बाद उसे expression में उपयोग किया जाता है।
- जबकि Post-Increment में पहले variable की value को expression में उपयोग किया जाता है, और उसके बाद उसकी value को increment किया जाता है।

Syntax:-

```
// PREFIX ++v
// POSTFIX v++
where v is a variable
 #include <stdio.h>
int increment(int a, int b)
{
a = 5;
// POSTFIX
```

Offline Coaching
www.engineersacademy.org
Mob: 8094441777
ENGINEERS ACADEMY
Your GATEway to Professional Excellence
IES • GATE • PSUs • JTO • IAS • NET
Online Coaching:
www.nimbus.org.in
Mob: 7374000888