



INFUSION NOTES
WHEN ONLY THE BEST WILL DO

राजस्थान कम्प्यूटर अनुदेशक

(राजस्थान कर्मचारी चयन आयोग (RSMSSB))

(हिंदी माध्यम)



भाग – 4

कंप्यूटर अध्ययन (भाग – 2)

प्रस्तावना

प्रिय पाठकों, प्रस्तुत नोट्स “राजस्थान कंप्यूटर अनुदेशक (हिंदी माध्यम)” को एक विभिन्न अपने अपने विषयों में निपुण अध्यापकों एवं सहकर्मियों की टीम के द्वारा तैयार किया गया है / ये नोट्स पाठकों को राजस्थान कर्मचारी चयन आयोग द्वारा आयोजित करायी जाने वाली परीक्षा “राजस्थान कंप्यूटर अनुदेशक भर्ती परीक्षा ” में पूर्ण संभव मदद करेंगे /

अंततः सतर्क प्रयासों के बावजूद नोट्स में कुछ कमियों तथा त्रुटियों के रहने की संभावना हो सकती है / अतः आप सूचि पाठकों का सुझाव सादर आमंत्रित हैं

प्रकाशक:

INFUSION NOTES

जयपुर, 302029 (RAJASTHAN)

मो : 9887809083

ईमेल : contact@infusionnotes.com

वेबसाइट : <http://www.infusionnotes.com>

Online Order करें - <https://rb.gy/86fb4c>

WhatsApp करें - <https://wa.link/f64b07>

मूल्य : ₹

संस्करण : नवीनतम

क्र. सं.	अध्याय	पृष्ठ सं.
1.	प्रोग्रामिंग भाषाएँ • सी का परिचय • C++ • JAVA • OOPs • Dot Net • Python • AI • Machine Learning • ब्लॉक चैन • प्रिंसिपल एंड प्रोग्रामिंग टेक्निक्स • Integrated Development Environment	1-292

Chapter - 1

प्रोग्रामिंग भाषाएँ (Programming Languages)

सी का परिचय :

C क्या है ?

c programming language क्या है ये जानने से पहले हमें ये पता होना जरूरी है की computer programming language क्या होती है.

जैसे हम humans आपस में बातचीत करने के लिए हिंदी, इंग्लिश किसी अन्य language का उपयोग करते हैं ठीक ऐसे ही computer से communicate करने के लिए हम programming languages का use करते हैं.

जैसे humans language में कुछ set of rules होते हैं जिनका use करके हम आपस में communicate करते हैं ठीक ऐसे ही सभी computer programming languages में कुछ set of rules होते हैं जिन्हें हम syntax कहते हैं.

इसी programming syntax को follow करके हम computer से कोई specific task के लिए instructions देते हैं यानी computer programs बनाते हैं.

c भी एक structured-oriented (procedural), middle-level और बहुत ही popular computer programming language है, जिसका इस्तेमाल मूल रूप से कंप्यूटर के विभिन्न एप्लिकेशन सॉफ्टवेयर और सिस्टम सॉफ्टवेयर को विकसित करने के लिए किया जाता है।

C लैंग्वेज का विकास डेनिस रिची ने किया था। उसके बाद 1978 में डेनिस रिची और ब्रेन कर्निघन ने सी भाषा का पहला संस्करण, 'The C प्रोग्रामिंग लैंग्वेज' प्रकाशित किया। C भाषा को computer का मातृ भाषा भी कहा जाता है, उदाहरण के लिए C भाषा के बाद जो भी भाषा बनाई गई थी (Java, PHP, C#, या C++) सभी C भाषा से उत्पन्न हुई हैं।

C Programming को Structured या Procedural Language क्यों कहते हैं?

c_Programming को Structured या Procedural Language इसलिए कहते क्योंकि इसमें हम अपने complex program को छोटे-छोटे blocks (modules) में divide कर देते हैं और इन blocks को हम functions कहते हैं.

C Programming को Middle Level Language क्यों कहते हैं?

c language भी low-level language की तरह direct memory access कर सकती है. इसके अलावा C में high-

level language की तरह user friendly features भी होते हैं जैसे की functions.

Low level और high level यानी के दोनों के features होने की वजह से C programming को हम middle-level language कहते हैं.

C (सी) भाषा का इतिहास

C Language को 1978 में ब्रायन कर्निघन और डेनिस रिची द्वारा प्रकाशित किया गया था। अब तक C भाषा के कई संस्करण आ चुके हैं, जो हैं -

K & R - यह C भाषा की मूल भाषा है। इस वर्जन को 1978 में पेश किया गया था। और इस वर्जन में स्टैंडर्ड 1/10 लाइब्रेरी जैसा फंक्शन उपलब्ध था।

एनएसआई सी और आईएसओ सी (ANSI C and ISO C) - इस संस्करण को अमेरिकी राष्ट्रीय मानक संस्थान (एनएसआई) कहा जाता है और इसे 1989/1990 में अंतराष्ट्रीय मानकीकरण संगठन (आईएसओ) द्वारा प्रकाशित किया गया था।

C99 - यह संस्करण 1999 में प्रकाशित हुआ था। और इस संस्करण में कुछ नई सुविधाएँ जोड़ी गईं जैसे - इनलाइन फंक्शन, कई नए डेटा प्रकार, लंबे इंट (long int) आदि।

C11 - यह संस्करण 2011 में प्रकाशित हुआ था। और इस संस्करण में कुछ नई सुविधाएँ भी जोड़ी गईं जैसे - पुस्तकालय, जिसमें टाइप जेनेरिक मैक्रोज, अनाम (anonymous) संरचनाएँ और C भाषा का इतिहास (सारणीबद्ध रूप) शामिल हैं।

भाषा	वर्ष	द्वारा विकसित
एल्गो	1960	अंतराष्ट्रीय समूह
बीसीपीएल	1966	मार्टिन रिचर्ड
पारंपरिक सी	1969	केन थॉम्पसन
कश्मीर और आर सी	1972	डेनिस रिची
एनएसआई सी	1978	कर्निघन और डेनिस रिची
एनएसआई/आईएसओ सी	1989	एनएसआई समिति
C90	1990	आईएसओ समिति
सी99	1999	आईएसओ समिति
सी11	2011	मानकीकरण समिति
सी18	2018	मानकीकरण समिति

C की विशेषता

सुवाह्यता (portability) :

C भाषा में लिखे गए प्रोग्राम काफी पोर्टेबल होते हैं, मतलब C भाषा में लिखे गए प्रोग्राम को अलग-अलग मशीनों या पीसी पर बिना किसी बदलाव के आसानी से चलाया जा सकता है। कंपाइलर और प्रीप्रोसेसर विभिन्न पीसी पर चलाना संभव बनाते हैं।

शक्तिशाली प्रोग्रामिंग भाषा :

C प्रोग्रामिंग लैंग्वेज बहुत तेज और कुशल प्रोग्रामिंग लैंग्वेज है क्योंकि यह डेटा टाइप, फंक्शन और कंट्रोल स्टेटमेंट का उपयोग करती है।

सरल प्रोग्रामिंग भाषा :

C भाषा में अंग्रेजी जैसे कमांड होते हैं, जिससे प्रोग्रामर के लिए कोड लिखना और समझना बहुत आसान हो जाता है।

संरचित(structured) उन्मुख(Oriented) भाषा :

C लैंग्वेज स्ट्रक्चर्ड ओरिएंटेड प्रोग्रामिंग लैंग्वेज है। स्ट्रक्चर्ड ओरिएंटेड प्रोग्रामिंग लैंग्वेज कोड की जटिलता को कम करती है, जिससे प्रोग्राम में काफी स्पष्टता आती है।

संकलक आधारित(compiler based) :

C Language Compiler Based Programming Language है, इसका मतलब C Language Programs को बिना Compiling के एक्जीक्यूट या रन नहीं किया जा सकता है।

सिस्टम्स आधारित भाषा :

C भाषा सिस्टम्स आधारित प्रोग्रामिंग भाषा है। सिस्टम्स आधारित प्रोग्रामिंग भाषा वह भाषा है जो नियमों और विनियमों का बहुत सख्ती से पालन करती है जैसे सी भाषा, सी ++ भाषा, जावा भाषा इत्यादि। यदि कोई प्रोग्रामिंग भाषा नियमों और विनियमों का सख्ती से पालन नहीं करती है, तो ऐसी प्रोग्रामिंग भाषा को सिस्टम्स आधारित प्रोग्रामिंग भाषा कहा जाता है। एचटीएमएल की तरह।

पॉइंटर्स का कुशल उपयोग :

पॉइंटर एक वेरिएबल है जो दूसरे वेरिएबल के एड्रेस की ओर इशारा करता है।

मध्य स्तर की भाषा :

C लैंग्वेज एक मिडिल लेवल प्रोग्रामिंग लैंग्वेज है, मतलब इसमें लो लेवल प्रोग्रामिंग की जा सकती है और हाई लेवल प्रोग्रामिंग भी की जा सकती है, जिससे सी लैंग्वेज की मदद से एप्लीकेशन सॉफ्टवेयर और सिस्टम सॉफ्टवेयर दोनों को बहुत आसानी से बनाया जा सकता है।

केस सेंसिटिव :

<https://www.infusionnotes.com/>

C भाषा एक केस संवेदी प्रोग्रामिंग भाषा है। केस सेंसिटिव का मतलब हम एक उदाहरण से समझते हैं, अगर हम C भाषा में एक जगह इस तरह से प्रिंटफ लिखते हैं और दूसरी जगह PRINTF इस तरह लिखते हैं, तो दोनों का मतलब C भाषा में अलग होता है।

C भाषा का अनुप्रयोग(Application of C language)

- C भाषा का उपयोग कंप्यूटर एप्लिकेशन सॉफ्टवेयर जैसे डेटाबेस, स्प्रेड शीट आदि बनाने के लिए किया जाता है।
- एंबेडेड सॉफ्टवेयर लिखने के लिए C भाषा का उपयोग किया जाता है।
- ऑपरेटिंग सिस्टम जैसे सिस्टम सॉफ्टवेयर बनाने के लिए C भाषा का उपयोग किया जाता है।
- कंप्यूटर और मोबाइल गेम्स जैसे ग्राफिक्स से संबंधित एप्लिकेशन बनाना।
- यूनिक्स कर्नेल(Unix kernel) पूरी तरह से सी भाषा में विकसित किया गया है।
- C भाषा का उपयोग नेटवर्क ड्राइव्स और डिवाइस ड्राइवर बनाने के लिए किया जाता है।
- कंपाइलर बनाने के लिए C भाषा का प्रयोग किया जाता है। कंपाइलर हाई लेवल कोड को लो लेवल कोड या मशीन कोड में बदलता है।

आपके सिस्टम पर C की स्थापना

C भाषा के लिए कई कंपाइलर उपलब्ध हैं। आप कोई भी डाउनलोड कर सकते हैं। यहां, हम टर्बो सी ++ स्थापित करने जा रहे हैं। हम इसे C और C++ दोनों के लिए इस्तेमाल कर सकते हैं।

टर्बो सी सॉफ्टवेयर स्थापित करने के लिए, आपको निम्नलिखित चरणों का पालन करना होगा।

टर्बो सी++ ऑनलाइन डाउनलोड करें।

C ड्राइव के अंदर टर्बोक(turboc) डायरेक्टरी बनाएं और tc zip को c:\turboc निकालें।

Install.exe फ़ाइल पर डबल क्लिक करें और इंस्टॉलेशन शुरू हो जाएगा।

C प्रोग्राम को लिखने के लिये c:\TC\BIN के अंदर tc (शॉर्टकट सिंबल) एप्लिकेशन file पर क्लिक करें।

- ऑनलाइन टर्बो सी++ सॉफ्टवेयर डाउनलोड करें
- आप कई साइटों से टर्बो सी++ ऑनलाइन डाउनलोड कर सकते हैं।
- सी ड्राइव में टर्बो सी (turbo c) निर्देशिका बनाएं और tc3.zip निकालें
- C इंस्टॉलेशन के लिए, आपको C: ड्राइव के अंदर एक नई निर्देशिका टर्बो c बनाने की आवश्यकता है। अब tc3.zip file को c:\turboc directory से निकालें।
- स्थापना के लिए install.exe फ़ाइल पर डबल क्लिक करें, अब, c:\turboc के अंदर स्थित इंस्टॉल आइकन पर क्लिक करें और इंस्टॉलेशन प्रक्रिया पूरी हो जाएगी।

विंडोज 7 या विंडोज 8 में, यह एप्लिकेशन को अनदेखा करने और बंद करने के लिए एक डायलॉग ब्लॉक दिखाएगा क्योंकि फुलस्क्रीन मोड समर्थित (supported) नहीं है। इग्नोर बटन पर क्लिक करें।

पहला C कार्यक्रम

C भाषा का प्रोग्राम शुरू करने से पहले, आपको सी कंसोल में प्रोग्राम को लिखना, संकलित करना और चलाना सीखना होगा।

1. फर्स्ट सी प्रोग्राम कैसे बनाये :-

C का पहला प्रोग्राम बनाने से पहले आपको कुछ जरूरी बातों का ध्यान रखना होता है, जो निम्नलिखित हैं :-

- हेडर फ़ाइल को शामिल करना आवश्यक है।
- हेडर फ़ाइल को # टैग से शुरू करना होगा।
- आपके program में एक मुख्य function होना बहुत महत्वपूर्ण है।
- c में स्टेटमेंट लिखते समय बड़े और छोटे अक्षरों का ध्यान रखना होता है।
- c में लगभग सभी कथनों को अर्धविराम से समाप्त किया जाना चाहिए।
- सी कंसोल(C console) खोलें और निम्नलिखित कोड लिखें :-

- `#include<stdio.h>int main () {printf(" Welcome to INFUSION NOTES") ; return 0 ;}`

Output :-

Welcome to INFUSION NOTES #include<stdio.h> में मानक इनपुट आउटपुट लाइब्रेरी फ़ंक्शन शामिल हैं।

Printf () फ़ंक्शन को stdio.h.int main () में परिभाषित किया गया है

मुख्य () फ़ंक्शन c भाषा में प्रत्येक प्रोग्राम का प्रवेश बिंदु है।

C प्रोग्राम को कैसे कंपाइल और रन करें

C प्रोग्राम को संकलित करने और चलाने के 2 तरीके हैं, मेनू और शॉर्टकट द्वारा।

C प्रोग्राम आउटपुट

मेनू द्वारा

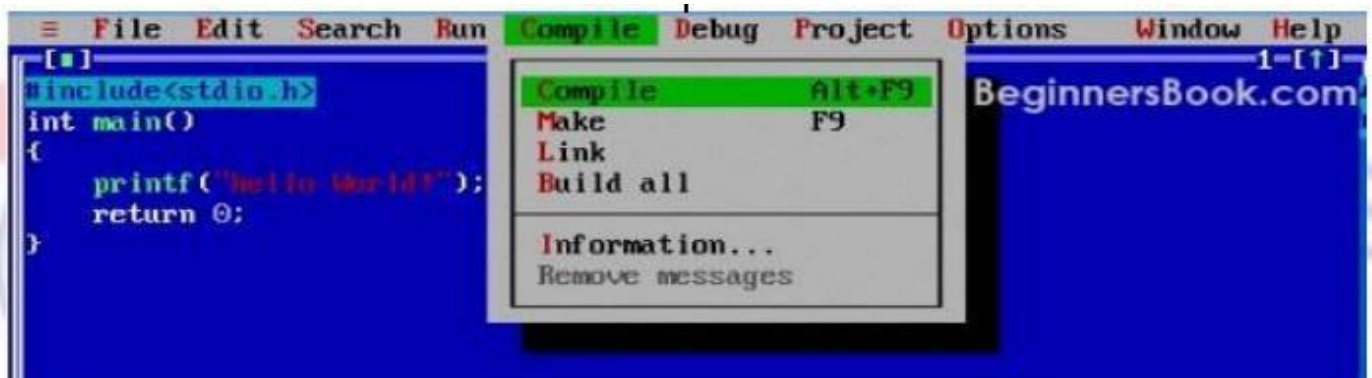
अब कंपाइल मेन्यू पर क्लिक करें और फिर सी प्रोग्राम को कंपाइल करने के लिए कंपाइल सब मेन्यू पर क्लिक करें। फिर रन मेन्यू पर क्लिक करें और फिर सी प्रोग्राम चलाने के लिए उप मेन्यू चलाएँ।



शॉर्टकट से

या, ctrl + f9 कुंजियाँ दबाएँ और प्रोग्राम को सीधे चलाएँ।

आप उपयोगकर्ता के स्क्रीन पर निम्न आउटपुट देखेंगे।



- आप उपयोगकर्ता स्क्रीन को किसी भी समय Alt + F5 कुंजी दबाकर देख सकते हैं।
- अब टर्बो c++ कंसोल पर वापस जाने के लिए Esc दबाएं।

फ्लोचार्ट का परिचय

फ्लो चार्ट हमारे कार्यक्रम का चित्रमय (graphical) प्रतिनिधित्व है। जब भी कोई C भाषा में प्रोग्राम लिखता है तो वह बिना फ्लो चार्ट के लिखता है। अगर हम अपने कार्यक्रम को समझना चाहते हैं, तो कार्यक्रम नहीं समझेगा। इसलिए हम अपने फ्लो चार्ट के माध्यम से प्रोग्राम को ग्राफिक रूप से प्रस्तुत कर सकते हैं। ताकि कोई भी प्रोग्राम के लॉजिक को आसानी से समझ सके।

जब हम किसी कंपनी में प्रोग्रामर के रूप में काम करते हैं फिर हमें कोडिंग से पहले फ्लोचार्ट बनाना होता है। हमने जो प्रोजेक्ट बनाया है वह आसानी से समझ में आता है और किसी और के द्वारा काम किया जाता है, इसलिए हमें फ्लो चार्ट बनाना होगा।

फ्लो चार्ट हमारे कार्यक्रम का गहन ज्ञान प्रदान करता है। यानी हम कोडिंग से पहले प्रोग्राम में आने वाली त्रुटियों और परिणामों को समझ सकते हैं

फ्लोचार्ट को पूरा करने के बाद हमें केवल फ्लोचार्ट के अनुसार कोडिंग लिखनी होती है। इससे समय की भी काफी बचत होती है

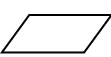
फ्लो चार्ट का सबसे बड़ा फायदा यह है। फ्लो चार्ट को देखकर आप किसी भी प्रोग्रामिंग लैंग्वेज में फ्लो चार्ट की प्रक्रिया को लागू कर सकते हैं। यानी आप अपना प्रोग्राम बना सकते हैं

फ्लोचार्ट हमारे कार्यक्रम का खाका (blue print) है। और फ्लो को चार्ट बनाने में भी काफी समय लगता है। एल्गोरिथम भी फ्लो चार्ट के अनुसार बनाया जाता है।

फ्लोचार्ट के प्रतीक

प्रोग्राम को आलेखीय रूप से निरूपित करने के लिए हमें कुछ प्रतीकों का उपयोग करना होता है। यह वह प्रतीक है जो हम प्रदान करते हैं। इन्हें फ्लो चार्ट सिंबल कहा जाता है।

प्रवाह चार्ट में प्रत्येक प्रतीक का एक अर्थ होता है जिसे उड़ता आसानी से समझ सकता है। नीचे फ्लो चार्ट सिंबल बताए जा रहे हैं :

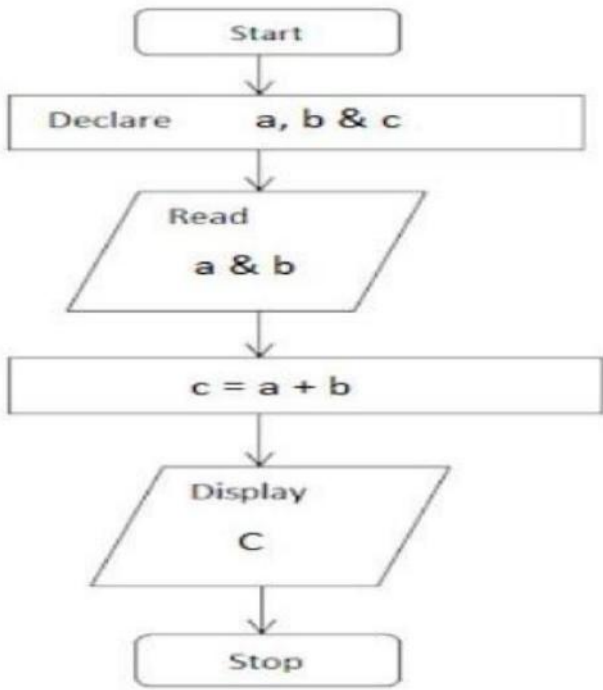
नाम	प्रतीक	व्याख्या
शुरु /अंत		इस चिन्ह को अंडाकार चिन्ह और टर्मिनल चिह्न भी कहा जाता है। इसका उपयोग फ्लो चार्ट को शुरू और समाप्त करने के लिए किया जाता है।
प्रसंस्करण		इस प्रतीक को प्रसंस्करण प्रतीक कहा जाता है। जिसका उपयोग प्रसंस्करण के लिए किया जाता है।
इनपुट आउटपुट		इस प्रतीक को समांतर चतुर्भुज प्रतीक कहा जाता है। इसका उपयोग इनपुट और आउटपुट प्रतीक को निरूपित करने के लिए किया जाता है।
फैसला		इसे हीरा प्रतीक कहते हैं। जिसका उपयोग निर्णय को निरूपित करने के लिए किया जाता है।
तीर		इसे तीर का प्रतीक कहा जाता है। इस चिन्ह का प्रयोग कार्यक्रम का प्रवाह को दिखाने के लिए किया जाता है

फ्लो चार्ट का उदाहरण

मान लीजिए कि हम एक अतिरिक्त प्रोग्राम करना चाहते हैं। जिसमें उपयोगकर्ता 2 नंबर दर्ज करेगा और उन दोनों नंबरों को जोड़कर परिणाम प्रदर्शित करना होगा। तो इसके लिए हम इस तरह फ्लो चार्ट बनाएंगे। जो नीचे दिया जा रहा है।

ऊपर का फ्लो चार्ट सबसे पहले अंडाकार चिन्ह वाले फ्लोचार्ट द्वारा शुरू किया जाता है। फिर 3 वैरिएबल्स a, b और c को रेक्टंगल सिंबल या प्रोसेसिंग सिंबल द्वारा घोषित किया जाता है।

फिर a और b पढ़ने की प्रक्रिया को समांतर चतुर्भुज प्रतीक द्वारा दिखाया गया है। फिर दोनों संख्याओं का योग आयत चिन्ह द्वारा c चर में किया जाता है।



फिर समानांतर चतुर्भुज द्वारा c चर(variable) प्रदर्शित करने की प्रक्रिया को दिखाया गया है। फ्लोचार्ट को तब टर्मिनल प्रतीक द्वारा रोक दिया जाता है।

कैरेक्टर (Character) सेट

वर्ण वे अक्षर होते हैं जिनका उपयोग किसी भी भाषा में शब्द बनाने के लिए किया जाता है। शब्दों से वाक्य बनते हैं, जो उस शब्द के अर्थ को प्रकट करते हैं।

कंप्यूटर भाषा एक ऐसी भाषा है जिसके उपयोग से हम कंप्यूटर को अपनी आवश्यकताओं के बारे में बता सकते हैं। इसलिए इस भाषा में भी अन्य भाषाओं की तरह व्याकरण है।

किसी भी भाषा को सीखने का सबसे पहला कदम उसके कैरेक्टर सेट को जानना होता है जिसे अंग्रेजी में कैरेक्टर सेट कहते हैं।

जैसे - जब हम कोई भाषा सीखते हैं तो सबसे पहले हमें उस भाषा का a, b, c, d पढ़ाया जाता है। ठीक उसी तरह C भाषा सीखने से पहले आपको इसके पात्रों के बारे में जानना होगा। भाषा में वर्ण सेट को चार भागों में बांटा गया है, जो इस प्रकार हैं।

अक्षर: अंग्रेजी भाषा के अक्षर A से Z और a से z इसके अक्षर हैं जिनका उपयोग प्रोग्राम लिखने के लिए किया जाता है। "सी" केस संवेदनशील भाषा है।

तो "सी" E और e दोनों को अलग-अलग अक्षरों के रूप में मानता है। इन अक्षरों को जोड़ने से पहचानकर्ता, कीवर्ड स्ट्रिंग आदि बनते हैं।

अंक: अंक का उपयोग सी भाषा में भी किया जाता है, जिसका उपयोग पहचानकर्ता संख्यात्मक मान के लिए किया जाता है। दस (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) हैं,

लेकिन आप उन्हें समूहबद्ध करके भी इसका उपयोग कर सकते हैं।

123,345,764 आदि।

विशेष वर्ण(special characters):

C भाषा में विशेष प्रकार के वर्ण होते हैं, जिनका प्रयोग केवल विशेष स्थानों में ही किया जाता है, आइए जानते हैं कि भाषा में किन विशेष वर्णों का प्रयोग किया जाता है।

C भाषा में विशेष वर्ण (special characters) कंप्यूटर विज्ञान की दृष्टि से महत्वपूर्ण होते हैं, क्योंकि इनका उपयोग कोडिंग के विभिन्न कार्यों के लिए किया जाता है। इन विशेष वर्णों का प्रयोग प्रोग्राम की संरचना, नियंत्रण, और डेटा हैंडलिंग के लिए किया जाता है। C भाषा में कुछ प्रमुख विशेष वर्ण निम्नलिखित हैं:

1. ; (सेमीकोलन)

कार्य: C में सेमीकोलन का उपयोग किसी स्टेटमेंट के अंत को सूचित करने के लिए किया जाता है।

उदाहरण:

`int a = 10; // स्टेटमेंट को समाप्त करता है`

2. {} (कल्लि ब्रेसिस)

कार्य: इनका उपयोग कोड के ब्लॉक (जैसे फंक्शन, लूप, और कंट्रोल स्टेटमेंट) को ग्रुप करने के लिए किया जाता है।

उदाहरण:

```

if (a > 0) {
    printf("Positive number");
}
  
```

3. () (पैरेंथेसिस)

कार्य: इनका उपयोग फंक्शन कॉल्स, कंट्रोल स्टेटमेंट्स और अभिव्यक्तियों में किया जाता है।

उदाहरण:

`printf("Hello, World!"); // फंक्शन कॉल`

4. [] (कोणीय ब्रेसिस)

कार्य: इसका उपयोग ऐरे (arrays) को परिभाषित और एक्सेस करने के लिए किया जाता है।

उदाहरण:

```

int arr[5]; // ऐरे का डिक्लेरेशन
arr[0] = 10; // ऐरे का एक्सेस
  
```

5. # (हैश)

कार्य: प्रीप्रोसेसर डायरेक्टिव्स (जैसे `#include`, `#define`) में इसका उपयोग किया जाता है।

उदाहरण:

`#include <stdio.h> // हेडर फाइल शामिल करना`

6. & (एम्पर्सैंड)

कार्य:

○ **एड्रेस ऑपरेटर:** इसका उपयोग किसी वैरिएबल के मेमोरी एड्रेस को प्राप्त करने के लिए किया जाता है।



- **लॉजिकल AND ऑपरेटर**: इसका उपयोग कंडीशन चेक करने के लिए किया जाता है।

• **उदाहरण:**

- `int a = 5;`
- `printf("%p", &a);` // मेमोरी एड्रेस प्रदर्शित करता है

7. ***** (एस्टेरिस्क)****

कार्य:

- **डेरिफरेंस ऑपरेटर (dereference operator)**: पॉइंटर से संदर्भित डेटा को एक्सेस करने के लिए इसका उपयोग किया जाता है।

- **गुणा ऑपरेटर (multiplication operator)**: इसका उपयोग गुणा करने के लिए किया जाता है।

• **उदाहरण:**

- `int a = 10, b = 20;`
- `int result = a * b;` // गुणा ऑपरेटर

8. **= (समतुल्यता/असाइनमेंट ऑपरेटर)**

- **कार्य:** इसका उपयोग किसी वेरिएबल को वैल्यू असाइन (assign) करने के लिए किया जाता है।

• **उदाहरण:**

- `int a = 5;` // असाइनमेंट ऑपरेटर

9. **== (दो समानता ऑपरेटर)**

- **कार्य:** इसका उपयोग किसी वेरिएबल की वैल्यू को चेक करने के लिए किया जाता है कि वह दूसरे वेरिएबल के बराबर है या नहीं।

• **उदाहरण:**

- `if (a == 10) {` // चेक करता है कि 'a' 10 के बराबर है या नहीं

- `printf("a is 10");`

- `}`

10. **!= (असमानता ऑपरेटर)**

- **कार्य:** इसका उपयोग दो मानों की असमानता को चेक करने के लिए किया जाता है।

• **उदाहरण:**

- `if (a != 10) {` // चेक करता है कि 'a' 10 के बराबर नहीं है

- `printf("a is not 10");`

- `}`

11. **++ (इन्क्रीमेंट ऑपरेटर)**

- **कार्य:** इसका उपयोग किसी वेरिएबल की वैल्यू को 1 से बढ़ाने के लिए किया जाता है।

• **उदाहरण:**

- `a++;` // 'a' की वैल्यू 1 बढ़ाएगा

12. **(डीक्रिमेंट ऑपरेटर)**

- **कार्य:** इसका उपयोग किसी वेरिएबल की वैल्यू को 1 से घटाने के लिए किया जाता है।

• **उदाहरण:**

- `a--;` // 'a' की वैल्यू 1 घटाएगा

13. **\ (बैकस्लैश)**

- **कार्य:** बैकस्लैश का उपयोग विशेष वर्णों (escape sequences) को प्रदर्शित करने के लिए किया जाता है, जैसे न्यू लाइन (\n), टैब (\t), बैकस्लैश (\), आदि।

• **उदाहरण:**

- `printf("Hello\nWorld");` // \n नया लाइन जोड़ेगा

14. **// (लाइन टिप्पणी)**

- **कार्य:** इसका उपयोग एकल लाइन टिप्पणी (single-line comment) लिखने के लिए किया जाता है।

• **उदाहरण:**

- `// यह एक टिप्पणी है`

15. **// (ब्लॉक टिप्पणी)**

- **कार्य:** इसका उपयोग मल्टी-लाइन टिप्पणी (multi-line comment) लिखने के लिए किया जाता है।

• **उदाहरण:**

- `/* यह एक`
- `मल्टी-लाइन`
- `टिप्पणी है */`

16. **' (सिंगल कोट)**

- **कार्य:** इसका उपयोग एकल अक्षर (character literal) को परिभाषित करने के लिए किया जाता है।

• **उदाहरण:**

- `char c = 'A';` // एकल अक्षर 'A' को परिभाषित करना

17. **" (डबल कोट)**

- **कार्य:** इसका उपयोग स्ट्रिंग लिटरल (string literal) को परिभाषित करने के लिए किया जाता है।

• **उदाहरण:**

- `char str[] = "Hello";` // स्ट्रिंग लिटरल "Hello" को परिभाषित करना

18. **, (कामा)**

- **कार्य:** इसका उपयोग दो या दो से अधिक तत्वों के बीच विभाजक के रूप में किया जाता है।

• **उदाहरण:**

- `int a = 10, b = 20;` // दो वेरिएबल्स को एक ही लाइन में परिभाषित करना

19. **.. (डॉट)**

- **कार्य:** इसका उपयोग स्ट्रक्चर (structure) के सदस्य तक पहुंचने के लिए किया जाता है।

• **उदाहरण:**

- `struct Point {`

- `int x;`

- `int y;`

- `};`

- `struct Point p;`

- `p.x = 10;` // डॉट ऑपरेटर के द्वारा 'x' तक पहुंचना

20. **-> (एरो ऑपरेटर)**

- **कार्य:** इसका उपयोग प्वाइंटर द्वारा संरचना (structure) के सदस्य तक पहुंचने के लिए किया जाता है।
- **उदाहरण:**

```
struct Point {
    int x;
    int y;
};
struct Point *p;
p->x = 10; // एरो ऑपरेटर का उपयोग
```

 ये विशेष वर्ण C भाषा में महत्वपूर्ण कार्यों के लिए उपयोग किए जाते हैं और प्रोग्रामिंग के विभिन्न पहलुओं में इनका व्यापक उपयोग होता है।

चर(variable)

- C भाषा में, जब हम अपने प्रोग्राम में कुछ डेटा वैल्यू का उपयोग करना चाहते हैं, तो हम इसे मेमोरी स्पेस में स्टोर कर सकते हैं और मेमोरी स्पेस को नाम दे सकते हैं ताकि इसे एक्सेस करना आसान हो जाए।
- किसी पते का नामकरण **चर** के रूप में जाना जाता है। वेरिएबल मेमोरी लोकेशन का नाम है।
- स्थिरांक के विपरीत, चर परिवर्तनशील होते हैं, हम किसी प्रोग्राम के निष्पादन के दौरान एक चर के मान को बदल सकते हैं।
- एक प्रोग्रामर एक सार्थक चर नाम चुन सकता है। उदाहरण : औसत, ऊंचाई, आयु, कुल आदि।

वेरिएबल का डेटाटाइप

C भाषा में एक वेरिएबल को एक प्रकार दिया जाना चाहिए, जो परिभाषित करता है कि वेरिएबल किस प्रकार का डेटा रखेगा।

char(चर): इसमें किसी कैरेक्टर को होल्ड/स्टोर कर सकते हैं।

int: एक पूर्णांक धारण करने के लिए प्रयुक्त।

फ्लोट: फ्लोट वैल्यू रखने के लिए प्रयुक्त होता है।

डबल: डबल वैल्यू रखने के लिए इस्तेमाल किया जाता है।

शून्य (void)

वेरिएबल नाम रखने के नियम

- Variable नाम एक अंक से शुरू नहीं होना चाहिए।
- Variable नाम में अक्षर, अंक और विशेष प्रतीक जैसे अंडरस्कोर _ शामिल हो सकते हैं।
- चर नाम में रिक्त या रिक्त स्थान की अनुमति नहीं है।
- चर नाम के रूप में खोजशब्दों(keywords) की अनुमति नहीं है।
- अपर और लोअर केस नामों को अलग-अलग माना जाता है, क्योंकि C केस-संवेदी है, इसलिए यह सुझाव दिया जाता है कि वेरिएबल नामों को लोअर केस में रखें।

- एक चर घोषित करना, परिभाषित करना और आरंभ करना।
- Program में उपयोग किए जाने से पहले चरों की घोषणा की जानी चाहिए। घोषणा निम्नलिखित बातें करती हैं।
 - यह संकलक को बताता है कि चर नाम क्या है
 - यह निर्दिष्ट करता है कि चर किस प्रकार का डेटा धारण करेगा।

जब तक वेरिएबल को परिभाषित नहीं किया जाता है, तब तक कंपाइलर को वेरिएबल में मेमोरी स्पेस आवंटित करने के बारे में चिंता करने की जरूरत नहीं है।

घोषणा संकलक को सूचित करने की तरह है कि कार्यक्रम में उपयोग किए जाने वाले डेटाटाइप के साथ एक चर मौजूद है।

मुख्य () फंक्शन के बाहर, बाहरी कीवर्ड का उपयोग करके एक चर घोषित किया जाता है।

Expression :

Expression चर, स्थिरांक, ऑपरेटरों और फंक्शन कॉल का संयोजन है।

यह अंकगणित, तार्किक और संबंधपरक हो सकता है, उदाहरण:-

`int z = x + y // अंकगणितीय व्यंजक`
`a > b // relational(संबंधपरक)`

`a == b // तार्किक(logical)`

`func(a, b) // फंक्शन कॉल`

Expression(व्यंजक) जो पूरी तरह से स्थिर मान से बने होते हैं, स्थिर(constant) व्यंजक(expression) कहलाते हैं।

अतः व्यंजक $121 + 17 - 110$ एक अचर व्यंजक है क्योंकि व्यंजक का प्रत्येक पद अचर है, लेकिन अगर एक पूर्णांक चर घोषित किया गया, तो अभिव्यक्ति $180 + 2 - j$ एक निरंतर expression का प्रतिनिधित्व नहीं करेगा।

C में कीवर्ड (keywords in C)

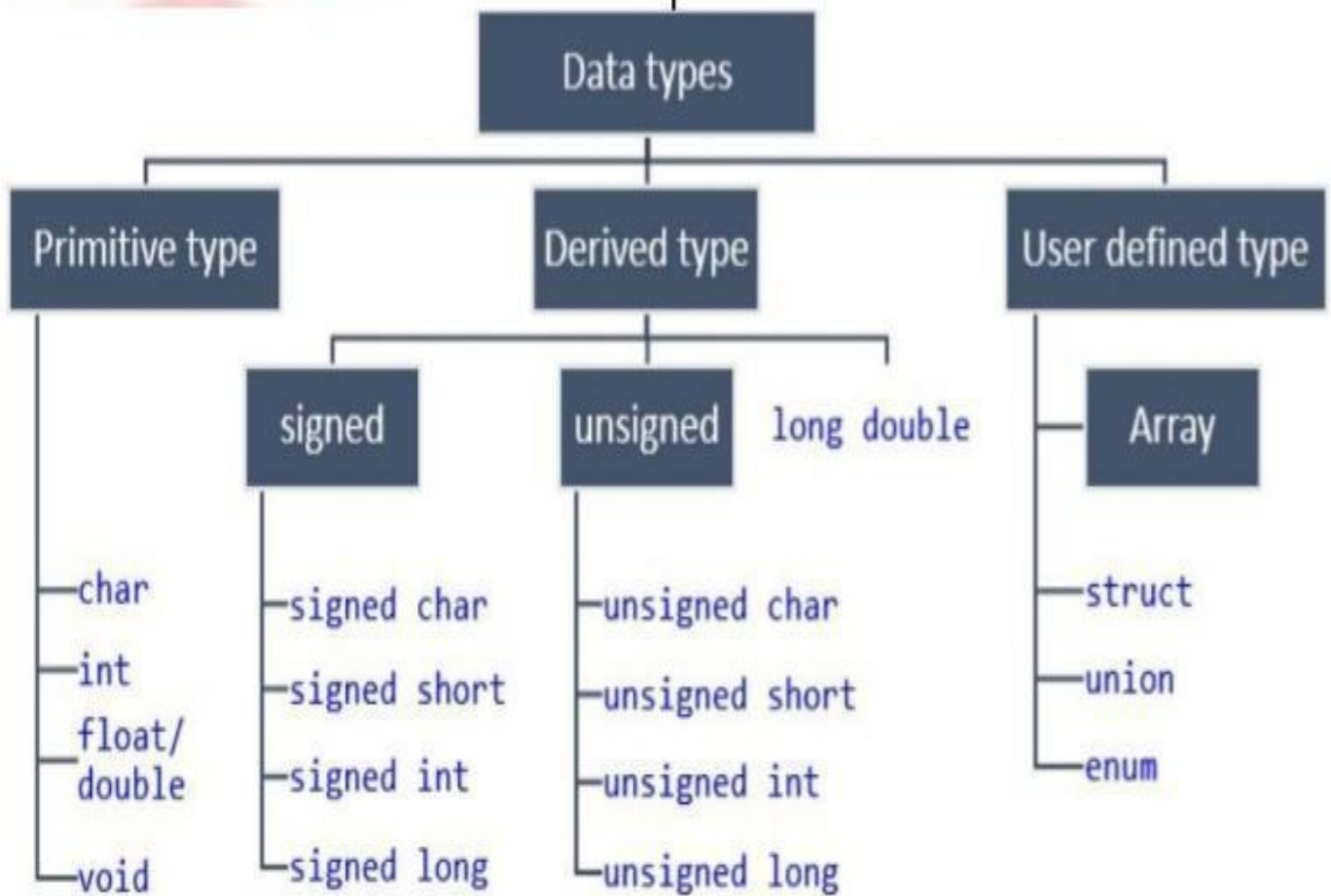
जैसे हमारी भाषा हिंदी में 'हंसो, रोओ, भागो' शब्दों का पहले से ही एक अर्थ है और इसलिए हम इन शब्दों का उपयोग किसी अन्य काम के लिए नहीं करते हैं, ठीक वैसे ही प्रोग्रामिंग में कीवर्ड(keywords) होते हैं। कीवर्ड आरक्षित शब्द हैं जिनका C प्रोग्रामिंग में पहले से ही विशिष्ट अर्थ है। अब जैसा कि हम समझ चुके हैं कि C Keywords का अर्थ और उपयोग पहले से ही निश्चित है, इसलिए हम किसी अन्य कार्य के लिए Keywords का उपयोग नहीं कर सकते हैं।

C प्रोग्रामिंग भाषा में 32 कीवर्ड होते हैं और C कीवर्ड हमेशा लोअरकेस(lowercase) में लिखे जाते हैं, सभी 32 कीवर्ड की सूची नीचे दी गई है :-



Data types

4. unsigned



5. long

C language में मुख्य रूप से तीन प्रकार के data type पाए जाते हैं।

- int [integer (पूर्णांक)]
- फ्लोट(float)
- चर (variable)

C भाषा में Data Types के प्रकार -

C language में मुख्य रूप से तीन प्रकार के data type होते हैं -

1. (Predefined)पूर्व-परिभाषित (Primitive डेटा) data types
2. Derived Data types
3. उपयोगकर्ता - परिभाषित डेटा प्रकार (user defined data types)

1. आदिम डेटा प्रकार (Primitive data types)

ये डेटा प्रकार अंतर्निहित (built in) या पूर्वनिर्धारित डेटा प्रकार हैं और उपयोगकर्ता द्वारा सीधे चर घोषित करने के लिए उपयोग किया जा सकता है। पूर्व-निर्धारित डेटा प्रकारों के उदाहरण

जब भी हम C language में कोडिंग करते हैं हमें वेरिएबल(variable) बनाने होंगे और इन वेरिएबल्स के प्रकार को **डेटा टाइप** कहा जाता है।

हम डेटा प्रकार घोषित किए बिना कोई प्रोग्राम नहीं बना सकते हैं, चाहे वह C भाषा हो या C++।

अगर हम किसी भी प्रकार का प्रोग्राम C language में बनाना चाहते हैं तो इसके लिए हमें variable बनाने होंगे।

डेटाटाइप संशोधक(Data type modifiers)

जैसा कि नाम का तात्पर्य है, डेटाटाइप संशोधक का उपयोग डेटा की लंबाई को संशोधित करने के लिए अंतर्निहित डेटा प्रकारों के साथ किया जाता है जो एक विशेष डेटा प्रकार धारण कर सकता है।

संशोधक(modifiers) C में कीवर्ड होते हैं जो C में मूल डेटा प्रकार का अर्थ बदलते हैं।

यह एक वेरिएबल के लिए आवंटित किए जाने वाले मेमोरी स्पेस की मात्रा को निर्दिष्ट करता है।

एक चर(variable) के लिए आवंटित स्मृति को संशोधित करने के लिए संशोधक मूल डेटा प्रकारों के साथ उपसर्ग करते हैं।

C प्रोग्रामिंग भाषा में 5 प्रकार के Data type modifiers हैं

1. Long
2. short
3. signed

यदि आप इस मान से अधिक के डेटा को वैरिएबल में स्टोर करना चाहते हैं तो आप "long int" कीवर्ड का उपयोग करके वैरिएबल बना सकते हैं। इस कीवर्ड की मदद से

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 byte	-32,768 to 32,767
signed short	2 byte	-32,768 to 32,767
unsigned short	2 byte	0 to 65,535
int	2 byte	-32,768 to 32,767
signed int	2 byte	-32,768 to 32,767
unsigned int	2 byte	0 to 65,535
short int	2 byte	-32,768 to 32,767
signed short int	2 byte	-32,768 to 32,767
unsigned short int	2 byte	0 to 65,535
long int	4 byte	-2,147,483,648 to 2,147,483,647
signed long int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned long int	4 byte	0 to 4,294,967,295
float	4 byte	1.2E-38 to 3.4E+38
double	8 byte	2.3E-308 to 1.7E+308
long double	10 byte	3.4E-4932 to 1.1E+4932

Integer data type:-

Integer प्रकार का वैरिएबल बनाने के लिए "int" कीवर्ड का उपयोग किया जाता है।

हम पूर्णांक प्रकार के चर में संख्यात्मक मान(numeric value) संग्रहीत करते हैं।

"int" डेटा प्रकार से बना वैरिएबल 2 बाइट, 4 बाइट और 8 बाइट डेटा स्टोर कर सकता है। लेकिन यह बात कंपाइलर पर निर्भर करती है।

कुछ कंपाइलर इंटर डेटा टाइप से बने वैरिएबल में 2 बाइट्स डेटा स्टोर करते हैं, जबकि कुछ 4 बाइट्स डेटा स्टोर करते हैं।

int (2 बाइट) -32,768 से 32,767 तक के मानों को स्टोर कर सकता है।

int (4 बाइट) - 2,147,483,648 से 2,147,483,647 तक के मानों को स्टोर कर सकता है।

बनाया गया वैरिएबल सामान्य वैरिएबल की तुलना में अधिक मात्रा में डेटा स्टोर कर सकता है।

%d फॉर्मेट स्पेसिफायर का उपयोग इंटर डेटा टाइप से बने वैरिएबल के मान को प्रिंट करने के लिए किया जाता है।

नोट :-

हम इंटर डेटा टाइप की मदद से बनाए गए वैरिएबल में दशमलव मान को स्टोर नहीं करते हैं।

यदि हम दशमलव मान को int डेटा प्रकार से बने चर में संग्रहीत करते हैं, तो उस मान का दशमलव बिंदु हट जाएगा और केवल पूर्ण संख्या संग्रहीत की जाएगी।

यदि हम दशमलव मान को स्टोर करना चाहते हैं तो हमें फ्लोट और डबल डेटा प्रकार का उपयोग करना होगा।

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:-

हम कैरेक्टर टाइप के डेटा को कैरेक्टर टाइप से बने वेरिएबल में स्टोर करते हैं।

"char" कीवर्ड का उपयोग कैरेक्टर टाइप का वेरिएबल बनाने के लिए किया जाता है।

कीवर्ड "चार" से बना वेरिएबल एक सिंगल कैरेक्टर को स्टोर करता है।

कैरेक्टर टाइप से बने वेरिएबल का साइज 1 बाइट होता है।

%c फॉर्मेट स्पेसिफायर का उपयोग चार डेटा प्रकार से बने वेरिएबल के मान को प्रिंट 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

फ्लोटिंग पॉइंट डेटा प्रकार

फ्लोटिंग पॉइंट डेटा प्रकार :-

• फ्लोट

• डबल फ्लोट

इसका उपयोग दशमलव बिंदु डेटा को स्टोर करने के लिए किया जाता है।

"float" कीवर्ड का उपयोग फ्लोटिंग पॉइंट प्रकार का वेरिएबल बनाने के लिए किया जाता है।

फ्लोट टाइप से बने एक वेरिएबल का आकार 4 बाइट्स होता है।

फ्लोट डेटा प्रकार से बने चर के मान को प्रिंट करने के लिए %f प्रारूप विनिर्देशक का उपयोग 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

1. Double data type:-

इसका उपयोग दशमलव बिंदु डेटा जैसे फ्लोट डेटा प्रकार को संग्रहीत करने के लिए भी किया जाता है।

डबल डेटा प्रकार का एक चर बनाने के लिए "double" कीवर्ड का उपयोग किया जाता है।

डबल टाइप से बने वेरिएबल का साइज 8 बाइट्स होता है।

। % lf प्रारूप विनिर्देशक का उपयोग दोहरे डेटा प्रकार से बने चर के मान को मुद्रित 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

डेटा प्रकार न केवल आदिम(primitive) डेटा प्रकार जैसे पूर्णांक या फ्लोटिंग डेटा प्रकार हो सकते हैं, बल्कि पॉइंटर डेटा प्रकार और यौगिक(compound) डेटा प्रकार जैसे यूनियन और स्ट्रक्चर(structs) भी हो सकते हैं।

C में sizeof () ऑपरेटर की आवश्यकता

मुख्य रूप से, प्रोग्राम आदिम डेटा प्रकारों के भंडारण आकार को जानते हैं। हालांकि डेटा प्रकार का भंडारण आकार स्थिर होता है, लेकिन विभिन्न प्लेटफार्मों में लागू होने पर यह भिन्न होता है। उदाहरण के लिए, हम sizeof() ऑपरेटर का उपयोग करके गतिशील रूप से सरणी स्थान आवंटित करते हैं:

```
int *ptr=malloc ( 10*sizeof(int) ) ;
```

उपरोक्त उदाहरण में, हम sizeof() ऑपरेटर का उपयोग करते हैं, जो कि int . के कास्ट type पर लागू होता है। हम मेमोरी आवंटित करने के लिए malloc () फंक्शन का उपयोग करते हैं और पॉइंटर लौटाते हैं जो इस आवंटित मेमोरी को इंगित(points) करता है।

मेमोरी स्पेस इंट डेटा प्रकार के कब्जे वाले बाइट्स की संख्या के बराबर है और इसे 10 से गुणा किया जाता है। The sizeof() ऑपरेटर ऑपरेंड के प्रकार के अनुसार अलग- अलग बताव करती है ।

C प्रोग्रामिंग में ऑपरेटर वरीयता और संबद्धता तालिका:

C प्रोग्रामिंग में ऑपरेटर वरीयता (Operator Precedence) और संबद्धता (Associativity) तालिका महत्वपूर्ण हैं, क्योंकि ये यह निर्धारित करती हैं कि किसी जटिल अभिव्यक्ति में विभिन्न ऑपरेटरों का क्रम और कार्यान्वयन कैसे होगा।

ऑपरेटर वरीयता (Operator Precedence):

ऑपरेटर वरीयता यह निर्धारित करती है कि यदि एक अभिव्यक्ति में कई ऑपरेटर हों, तो उन्हें किस क्रम में निष्पादित किया जाएगा। उच्च वरीयता वाले ऑपरेटर पहले निष्पादित होंगे। C में ऑपरेटर वरीयता कुछ इस प्रकार से होती है:

C में ऑपरेटर वरीयता और संबद्धता तालिका:

क्रम	ऑपरेटर	विवरण	संबद्धता
1	[] -> .	पैरेंटिसिस, ऐरे इंडेक्सिंग, स्ट्रक्चर और यूनियन में सदस्य, प्वाइंटर डिरेक्ट एक्सेस	बायाँ से दायें (Left to Right)
2	++ -- (प्रत्येक)	इन्क्रिमेंट / डिक्रीमेंट (प्रारंभिक और पश्चात)	दायें से बायें (Right to Left)
3	! ~ + - & * sizeof	युनरी ऑपरेटर (नकारात्मकता, बिटवाइज NOT, पॉजिटिव/निगेटिव, प्वाइंटर, आकार)	दायें से बायें (Right to Left)
4	* / %	गुणा, भाग, और शेषफल	बायाँ से दायें (Left to Right)
5	+ -	जोड़ और घटाव	बायाँ से दायें (Left to Right)
6	<< >>	बिट शिफ्ट ऑपरेटर	बायाँ से दायें (Left to Right)
7	< <= > >=	तुलना ऑपरेटर	बायाँ से दायें (Left to Right)
8	== !=	समानता और असमानता ऑपरेटर	बायाँ से दायें (Left to Right)
9	&	बिटवाइज AND	बायाँ से दायें (Left to Right)
10	^	बिटवाइज XOR	बायाँ से दायें (Left to Right)
11		बिटवाइज OR	बायाँ से दायें (Left to Right)
12	&&	लॉजिकल AND	बायाँ से दायें (Left to Right)
13		लॉजिकल OR	बायाँ से दायें (Left to Right)
14	? :	टर्नरी ऑपरेटर	बायाँ से दायें (Left to Right)
15	= += -= *= /= %= &= ^= `	= <<= >>= `	असाइनमेंट ऑपरेटर
16	,	कॉमा ऑपरेटर	बायाँ से दायें (Left to Right)

ऑपरेटर वरीयता का उदाहरण:

मान लीजिए आपके पास निम्नलिखित अभिव्यक्ति है:

c

```
int result = 3 + 4 * 2;
```

यहां, गुणा (*) ऑपरेटर की वरीयता जोड़ (+) ऑपरेटर से अधिक है। इसलिए, सबसे पहले 4 * 2 होगा, और फिर 3 + 8 किया जाएगा, जिसका परिणाम 11 होगा।

संबद्धता (Associativity):

C++ Language दुनिया की टॉप 10 प्रोग्रामिंग लैंग्वेज में से एक है। हमें इसे stackoverflow survey और TIOBE Index के अनुसार सी++ लैंग्वेज दुनिया की सबसे पॉपुलर प्रोग्रामिंग लैंग्वेज है।

Introduction of C++

C++ एक object-oriented programming language है जिसे Bjarne Stroustrup ने 1983 में AT & Bell Laboratories में विकसित किया था।

C++ language, सी लैंग्वेज की तरह ही है। C Language में बनाये गए सभी प्रोग्राम हम बिना कुछ बदलाव या फिर मामूली से बदलाव के साथ C++ language में बड़े ही आसानी से बना सकते हैं।

हम कह सकते हैं कि C++ language, सी लैंग्वेज का Superset है जिसमें सी लैंग्वेज के फीचर्स तो हैं ही साथ में इसके कुछ और एडिशनल फीचर्स भी हैं।

C++ language भी सी लैंग्वेज की तरह एक Middle level प्रोग्रामिंग लैंग्वेज है जिसमें हम low level और high level दोनों ही तरह के प्रोग्राम्स बड़े ही आसानी से बना सकते हैं। C++ language की इस फीचर के कारण इसकी मदद से Application software तथा System software दोनों ही तरह के सॉफ्टवेयर आसानी से बना सकते हैं।

C++ लैंग्वेज का इतिहास (History of C++)

Bjarne Stroustrup ने C++ language पर 1978 में काम करना शुरू किया और 1979 में इसका पहला वर्शन रिलीज़ किया। शुरुवात में इनके अनुसार ये कोई नई प्रोग्रामिंग लैंग्वेज नहीं थी इनका कहना था कि ये सी लैंग्वेज का ही एक Extended वर्जन है।

जब Bjarne Stroustrup ने C++ language में Classes का concept जोड़ा तब इसे उस समय "C With Classes" कहा गया।

Bjarne Stroustrup ने Classes के साथ साथ कुछ और नए कांसेप्ट add किया जिसके चलते 1983 में Bjarne Stroustrup द्वारा बनाये गए लैंग्वेज का नाम C++ पड़ा। C++ में ये जो इंक्रीमेंट ऑपरेटर "++" है वो इंडिकेट करता है कि C++ language, सी लैंग्वेज का Extended वर्जन है।

C++ language को 1998 में ISO committee द्वारा standardized किया गया और सन 2003 में इसका छोटा सा अपडेट रिलीज़ किया गया जिसके चलते इसे C++03 कहा गया।

2011, 2014, और 2017 में C++ language के नए स्टैंडर्ड्स रिलीज़ किया गया जिसके चलते C++ language को C++11, C++14, और C++17 कहा गया।

C++ language का हर नया standard एक नए फीचर्स के साथ आता है जो प्रोग्रामर को प्रोग्रामिंग में काफी मदद करता है।

Language—Year

C With
Classes —1979

C++ —1983

C++03 — 2003

C++11 — 2011

C++14 — 2014

C++17 — 2017

C++20 — 2020

C++ लैंग्वेज की विशेषता

- Simple
- Machine Independent or Portable
- Mid-level programming language
- Structured programming language
- Rich Library
- Memory Management
- Fast Speed
- Pointers
- Recursion
- Extensible
- Object-Oriented
- Compiler based
- Statically typed

1) simple

C++ इस अर्थ में एक सरल भाषा है क्योंकि यह संरचित दृष्टिकोण (भागों में समस्या को तोड़ने के लिए), library functions का समृद्ध सेट, डेटा संग्रह आदि प्रदान करता है।

2) मशीन स्वतंत्र या पोर्टेबल

असेंबली लैंग्वेज के विपरीत, C प्रोग्राम्स को कई मशीनों में बहुत कम या कोई बदलाव किए बिना चलाया जा सकता है। लेकिन यह प्लेटफॉर्म-स्वतंत्र नहीं है।

3) मध्य-स्तरीय प्रोग्रामिंग भाषा

C ++ का उपयोग निम्न स्तर की प्रोग्रामिंग करने के लिए भी किया जाता है। इसका उपयोग सिस्टम एप्लिकेशन जैसे कि कर्नेल, ड्राइवर आदि को विकसित करने के लिए किया जाता है। यह उच्च स्तरीय भाषा की सुविधा का भी समर्थन करता है। इसीलिए इसे मध्य-स्तरीय भाषा के रूप में जाना जाता है।

4) संरचित प्रोग्रामिंग भाषा

C ++ एक संरचित प्रोग्रामिंग भाषा है इस अर्थ में कि हम फंक्शन का उपयोग करके प्रोग्राम को भागों में तोड़ सकते हैं। इसलिए, इसे समझना और संशोधित करना आसान है।

5) रिच लाइब्रेरी

C ++ बहुत सारे इनबिल्ट फंक्शंस प्रदान करता है जो विकास को तेज़ बनाता है।

6) मेमोरी प्रबंधन

यह **dynamic** मेमोरी आवंटन की सुविधा का समर्थन करता है। **C ++** लैंग्वेज में, हम **free()** फंक्शन को कॉल करके किसी भी समय आवंटित मेमोरी को फ्री कर सकते हैं।

7) गति

C ++ भाषा का संकलन और निष्पादन समय तेज़ है।

8) pointers

C ++ पॉइंटर्स की सुविधा प्रदान करता है। हम सीधे पॉइंटर्स का उपयोग करके मेमोरी के साथ इंटरैक्ट कर सकते हैं। हम पॉइंटर्स का उपयोग मेमोरी, संरचनाओं, **functions**, **arrays** आदि के लिए कर सकते हैं।

9) recursion

C ++ में, हम फंक्शन को फंक्शन के भीतर कॉल कर सकते हैं। यह प्रत्येक फंक्शन के लिए कोड पुनः प्रयोज्य प्रदान करता है।

10) एक्स्टेंसिबल

C ++ भाषा एक्स्टेंसिबल है क्योंकि यह आसानी से नई सुविधाओं को अपना सकती है।

11) ऑब्जेक्ट ओरिएंटेड

C ++ ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग लैंग्वेज है। **OOP** विकास और रखरखाव को आसान बनाता है जहां **procedure oriented** प्रोग्रामिंग भाषा के रूप में यह प्रबंधन करना आसान नहीं है।

12) संकलक आधारित

C ++ एक कंपाइलर आधारित प्रोग्रामिंग भाषा है, इसका मतलब है कि संकलन के बिना कोई **C ++** प्रोग्राम निष्पादित नहीं किया जा सकता है। पहले हमें संकलक का उपयोग करके हमारे कार्यक्रम को संकलित करने की आवश्यकता है और फिर हम अपने कार्यक्रम को निष्पादित कर सकते हैं।

II. 13. Statically typed

C++ language एक **Statically typed** प्रोग्रामिंग लैंग्वेज है जिसमें वेरिएबल के डेटा टाइप को रन टाइम में न चेक करके कंपाइल टाइम में चेक किया जाता है।

C++ लैंग्वेज क्यों सीखें? (Why to Learn C++)

- **C++ लैंग्वेज** हार्डवेयर के काफी नजदीक होता है इसकी मदद से हम **Low Level** प्रोग्रामिंग कर सकते हैं। **C++ लैंग्वेज** की मदद से बने प्रोग्राम द्वारा हम मेमोरी मैनेजमेंट बड़े ही आसानी से कर सकते हैं।
- **C++ language** की मदद से आप ऑब्जेक्ट ओरिएंटेड कांसेप्ट को बड़े ही आसानी से सिख सकते हैं। ऑब्जेक्ट ओरिएंटेड कांसेप्ट आपके आगे की प्रोग्रामिंग यात्रा को आसान बना देता है।
- **C++** के द्वारा आप सिस्टम सॉफ्टवेयर और एप्लीकेशन सॉफ्टवेयर दोनों ही तरह के सॉफ्टवेयर बनाने में सक्षम हो जाते हैं।

Uses of C++

- **C ++** का उपयोग **system software** (जैसे **windows**, **Linux**, **mobile OS**) से एप्लीकेशन सॉफ्टवेयर (जैसे, **Photoshop**, **corel-draw**, **Microsoft-office**) बनाने में किया जाता है।
- **C ++** भाषा का उपयोग **Gaming क्षेत्र** में, **animation**, **motion graphics**, **virtual reality device** जैसे **high usage** वाले **software** विकसित करने में भी किया जाता है।

Introduction oops in C++

C ++, **C language** का ही **updated version** है लेकिन जब **C language** पहले से ही थी, तब **C ++** की आवश्यकता क्यों पड़ी। **C** और **C ++** के बीच मुख्य अंतर **OOPs** का है।

C एक **POP (Procedure Oriented Programming)** भाषा है जबकि **C ++** **Object Oriented Programming language** है हालांकि, **C ++ OOPs** और **POP** दोनों को **support** करती है जबकि **C** केवल **POP** को **support** करता है।

जब **OOPs features** को **C** में जोड़ा गया था, तो इसे **C++** नाम दिया गया। **OOP** में **programming** करने से पहले,

POP और OOPs के बारे में जान लेते हैं ताकि आपको पता चल जाए कि एक नई भाषा C++ की कमी क्यों महसूस हुई।

Object-Oriented Programming (OOPs)

OOP को Object Oriented Programming के नाम से जानते हैं।

OOP का सिद्धांत प्रथमतः 1967 में प्रकाश में आया। यह सिद्धांत 1980 में लोकप्रिय हुआ। आजकल यह तथ्य अधिकतर

Programming Language का भाग है।

सरल शब्दों में OOPs, collection of objects होता है। जिसमें program object में बंटा होता है जहाँ object data -member function member का collection होता है जो एक साथ एक set के रूप में logically एक दूसरे से related होते हैं, जिसे class कहा जाता है। इसलिए हम कह सकते हैं कि एक class OOPs का important feature होता है। ऑब्जेक्ट ओरिएण्टेड प्रोग्रामिंग (OOPS) में फंक्शन के बजाए डेटा को प्रोग्राम के विकास का महत्वपूर्ण भाग माना जाता है। इसमें डेटा उन फंक्शन के साथ ही जुड़े होते हैं जिन्हें उसे प्रयोग करना है।

इस प्रकार डेटा बाहरी Functions में प्रयोग तथा उनके द्वारा हो सकने वाले संभावित परिवर्तनों से बचा रहता है। OOPs में आप complex code को Object की मदद से स्मॉलर प्रॉब्लम में तोड़ सकते हो और हल कर सकते हो। यह सॉफ्टवेयर industries में अत्यंत उपयोगी concept है। जिसका उपयोग सॉफ्टवेयर को अत्यंत reliable और reusable बनाता है। OOP, data को प्रोग्राम के विकास से प्रयुक्त होने वाला एक महत्वपूर्ण घटक मानती है और इस data को system पर मुक्त रूप से एक स्थान से दूसरे स्थान पर ले जाने की अनुमति नहीं देती है। यह data को operate करने वाले function के निकट रखती है और बाह्य function के आकस्मिक संशोधनों से इसकी सुरक्षा करती है।

Features of Object Oriented Programming

Object Oriented Programming Language के निम्न Features हैं, और इसके Basic Concept निम्न हैं-

1. Objects
2. Class
3. Data Abstraction
4. Polymorphism
5. Data Encapsulation or Data Hiding
6. Inheritance
7. Message Passing
8. Reusability
9. Dynamic Binding

C++ में ऑब्जेक्ट-ओरिएण्टेड प्रोग्रामिंग (OOP) की प्रमुख अवधारणाओं को समझने के लिए नीचे दिए गए प्रत्येक विषय को हिंदी में विस्तार से समझाया गया है:

1. ऑब्जेक्ट (Objects)

ऑब्जेक्ट किसी क्लास का एक उदाहरण (instance) होता है। इसका मतलब है कि जब हम किसी क्लास को परिभाषित करते हैं, तो उससे वास्तविक दुनिया का एक ऑब्जेक्ट बनता है। एक ऑब्जेक्ट में डेटा और फंक्शन होते हैं जो उस ऑब्जेक्ट के व्यवहार और गुण को दर्शाते हैं।

उदाहरण:

```
class Car {
```

```
public:
```

```
    string model;
```

```
    int year;
```

```
    void start() {
```

```
        cout << "Car started" << endl;
```

```
    }
```

```
};
```

```
int main() {
```

```
    Car myCar; // 'myCar' एक ऑब्जेक्ट है
```

```
    myCar.model = "Toyota";
```

```
    myCar.year = 2022;
```

```
    myCar.start(); // 'start()' फंक्शन को कॉल करना
```

```
    return 0;
```

```
}
```

यहां, myCar एक ऑब्जेक्ट है जो Car क्लास का उदाहरण है।

2. क्लास (Class)

क्लास एक टेम्पलेट है, जिससे ऑब्जेक्ट बनते हैं। यह एक कस्टम डेटा टाइप है जो संबंधित डेटा (वेरिएबल्स) और कार्य (फंक्शन्स) को एक साथ एक इकाई में बांधता है। क्लास के अंदर डेटा मेंबर और फंक्शन मेंबर होते हैं।

उदाहरण:

```
class Car {
```

```
public:
```

```
    string model;
```

```
    int year;
```

```
    void start() {
```

```
        cout << "Car started" << endl;
```

```
    }
```

```
};
```

यहां Car एक क्लास है जो model, year और start() जैसे फंक्शन्स और डेटा को परिभाषित करता है।

3. डेटा एब्सट्रैक्शन (Data Abstraction)

डेटा एब्स्ट्रैक्शन का मतलब है केवल महत्वपूर्ण जानकारी को दिखाना और बाकी जानकारी को छुपाना। इसका मुख्य उद्देश्य जटिलता को कम करना है और केवल आवश्यक डेटा को उपयोगकर्ता तक पहुँचाना है। C++ में हम डेटा एब्स्ट्रैक्शन को **क्लास** और **मेम्बर फंक्शन्स** के माध्यम से करते हैं।

उदाहरण:

```
class Car {
private:
    string model;
public:
    void setModel(string m) {
        model = m;
    }
    string getModel() {
        return model;
    }
};
```

यहाँ, **model** डेटा को **private** किया गया है, जिससे वह सीधे एक्सेस नहीं किया जा सकता, और केवल **setModel()** और **getModel()** फंक्शन्स के माध्यम से उसे सेट और प्राप्त किया जा सकता है।

4. पॉलीमॉर्फिज़्म (Polymorphism)

पॉलीमॉर्फिज़्म का मतलब है "एक नाम से कई रूप"। इसका अर्थ है कि एक ही फंक्शन या ऑपरेटर का अलग-अलग प्रकार से व्यवहार हो सकता है। C++ में पॉलीमॉर्फिज़्म मुख्य रूप से **फंक्शन ओवरलोडिंग** और **ऑपरेटर ओवरलोडिंग** के रूप में पाया जाता है। इसके अलावा, **वर्चुअल फंक्शन्स** के माध्यम से रनटाइम पॉलीमॉर्फिज़्म भी होता है।

उदाहरण:

```
class Animal {
public:
    virtual void sound() {
        cout << "Animal makes a sound" << endl;
    }
};
```

```
class Dog : public Animal {
public:
    void sound() override {
        cout << "Dog barks" << endl;
    }
};
```

```
int main() {
    Animal* animal;
```

```
Dog dog;
animal = &dog;
animal->sound(); // Dog barks
return 0;
}
```

यहाँ, **sound()** फंक्शन अलग-अलग कक्षाओं में अलग-अलग कार्य करता है, यह रनटाइम पॉलीमॉर्फिज़्म का उदाहरण है।

5. डेटा इनकैप्सुलेशन या डेटा हाइडिंग (Data Encapsulation or Data Hiding)

डेटा इनकैप्सुलेशन का मतलब है डेटा को सुरक्षित रखना और उसे सीधे एक्सेस से छिपाना। C++ में हम डेटा हाइडिंग को **private** और **protected** एक्सेस मॉडिफायर के द्वारा करते हैं, जिससे डेटा केवल क्लास के अंदर ही एक्सेस किया जा सकता है और बाहरी कोड से उसे सुरक्षित रखा जा सकता है।

उदाहरण:

```
class Account {
private:
    double balance; // डेटा हाइडिंग
public:
    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
        }
    }
    double getBalance() {
        return balance;
    }
};
```

यहाँ, **balance** डेटा **private** है, जिसे सीधे एक्सेस नहीं किया जा सकता। इसके लिए **deposit()** और **getBalance()** जैसे फंक्शन्स दिए गए हैं।

6. इनहेरिटेंस (Inheritance)

इनहेरिटेंस का मतलब है कि एक क्लास दूसरी क्लास के गुणों (properties) और कार्यों (methods) को अपना सकती है। यह OOP का एक प्रमुख सिद्धांत है, जो कोड के पुनः प्रयोग (reusability) और वंशानुक्रम (hierarchy) को सक्षम बनाता है।

उदाहरण:

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

```
};
```

```
class Dog : public Animal {
```

```
public:
```

```
void bark() {
```

```
    cout << "Barking" << endl;
```

```
}
```

```
};
```

```
int main() {
```

```
    Dog dog;
```

```
    dog.eat(); // Animal class का method
```

```
    dog.bark(); // Dog class का method
```

```
    return 0;
```

```
}
```

यहां, Dog क्लास को Animal क्लास से गुण प्राप्त हुए हैं (इनहेरिटेंस)।

7. मैसेज पासिंग (Message Passing)

मैसेज पासिंग OOP में एक तरीका है जिसमें एक ऑब्जेक्ट दूसरे ऑब्जेक्ट को एक मैसेज भेजता है। C++ में यह आमतौर पर फंक्शन्स के माध्यम से होता है, जिसमें एक ऑब्जेक्ट दूसरे ऑब्जेक्ट के मेथड्स को कॉल करता है।

उदाहरण:

```
class Dog {
```

```
public:
```

```
void bark() {
```

```
    cout << "Barking" << endl;
```

```
}
```

```
};
```

```
int main() {
```

```
    Dog dog;
```

dog.bark(); // Dog ऑब्जेक्ट से 'bark()' को मैसेज भेजा जा रहा है

```
    return 0;
```

```
}
```

8. रीयूजेबिलिटी (Reusability)

रीयूजेबिलिटी का मतलब है कि एक बार लिखा गया कोड पुनः कई स्थानों पर उपयोग किया जा सकता है। C++ में इनहेरिटेंस और क्लासेस के माध्यम से हम कोड को पुनः उपयोग करने की सुविधा प्राप्त करते हैं।

उदाहरण:

```
class Animal {
```

```
public:
```

```
void eat() {
```

```
    cout << "Eating food" << endl;
```

```
}
```

```
};
```

```
class Dog : public Animal {
```

```
public:
```

```
void bark() {
```

```
    cout << "Barking" << endl;
```

```
}
```

```
};
```

यहां, Animal क्लास का कोड Dog क्लास में पुनः उपयोग हो रहा है (रीयूजेबिलिटी)।

10. डायनामिक बाइंडिंग (Dynamic Binding)

डायनामिक बाइंडिंग का मतलब है कि किसी फंक्शन का वास्तविक कार्य रनटाइम के दौरान निर्धारित होता है। C++ में यह **वर्चुअल फंक्शन्स** के द्वारा होता है, जहां एक बेस क्लास के प्वाइंटर्स से डेरिड क्लास के फंक्शन्स को कॉल किया जा सकता है।

उदाहरण:

```
class Base {
```

```
public:
```

```
virtual void show() {
```

```
    cout << "Base class" << endl;
```

```
}
```

```
};
```

```
class Derived : public Base {
```

```
public:
```

```
void show() override {
```

```
    cout << "Derived class" << endl;
```

```
}
```

```
};
```

```
int main() {
```

```
    Base* basePtr;
```

```
    Derived derivedObj;
```

```
    basePtr = &derivedObj;
```

```
    basePtr->show(); // Derived class
```

```
    return 0;
```

```
}
```

यहां, show() फंक्शन का चयन रनटाइम पर किया जा रहा है, जो डायनामिक बाइंडिंग का उदाहरण है।

निष्कर्ष:

C++ में ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग (OOP) के ये

सिद्धांत - ऑब्जेक्ट, क्लास, डेटा एब्स्ट्रैक्शन, पॉलीमॉर्फिज़्म, डेटा इनकैप्सुलेशन, इनहेरिटेंस, मैसेज पासिंग, रीयूजेबिलिटी, और डायनामिक बाइंडिंग - सॉफ्टवेयर डिजाइन को अधिक प्रभावी, संरचित और सुलभ बनाते हैं। इन सिद्धांतों का सही तरीके से उपयोग करने से बड़े और जटिल सिस्टमों को आसानी से प्रबंधित किया जा सकता है।

Advantage Of OOPS

Object Oriented Programming Language के निम्नलिखित लाभ हैं-

- इस पद्धति में Inheritance के द्वारा redundant code को हटाया जा सकता है एवं पहले से उपस्थित Classes के उपयोग को extend किया जा सकता है।
- डेटा हाइडिंग का Concept प्रोग्रामर को सुरक्षित प्रोग्राम बनाने में सहायता करता है जिसे कोड द्वारा प्रोग्राम के दूसरे भागों में प्रयोग नहीं किया जा सकता है।
- OOP की तकनीक से एक Program को Quickly (शीघ्रतापूर्वक) Operational हेतु Objects के समूहों के आधार पर अनेक भागों में विभक्त किया जा सकता है।
- इस तकनीक से छोटे - छोटे Program के Combination द्वारा Large Program सरलतापूर्वक तैयार किए जा सकते हैं।

को बार - बार नहीं लिखना पड़ता है, जिसके कारण प्रोग्राम में समय की बचत होती ही है तथा Develop की क्षमता में वृद्धि होती है।

- इस तकनीक के द्वारा Object की Classes के साथ उनसे संबंधित Functions को भी integrated कर दिया जाता है, जिससे प्रोसेसिंग का कार्य सुगम और सुरक्षित हो जाता है।
- Objects के बीच Communication के लिए मैसेज पासिंग Method द्वारा External System के साथ इंटरफेस करना आसान होता है।
- इस प्रोग्रामिंग के द्वारा सॉफ्टवेयर जटिलता (complexity) को आसानी से हल किया जा सकता है।
- सॉफ्टवेयर विकसित (develop) करना आसान होता है।

Keywords.

C++ में keywords ऐसे reserved words हैं जिनकी meaning पहले से ही define की गई है तथा इनके अपने कुछ specific features हैं, इनकी मीनिंग को change नहीं किया जा सकता है। C++ में 32 keywords हैं।

C++ Reserved Words

Asm	double	new	switch
Auto	else	operator	template
break	enum	private	this
Case	extern	protected	throw
catch	float	public	try
Char	for	register	typedef
class	friend	return	union
const	goto	short	unsigned
continue	if	signed	virtual
default	inline	sizeof	void
delete	int	static	volatile
Do	long	struct	while

- इसके द्वारा कार्यरत स्तरीय मॉड्यूलों का संबंध सरलतापूर्वक दूसरे मॉड्यूलों से हो जाता है, जिससे कोड
- <https://www.infusionnotes.com/>

Basics usage of these keywords -

Example for Run Package

java pack2.B

Compile and Run Package

यह उपर्युक्त पैकेजों का आउटपुट है। यहाँ '-d' इस फ़ोल्डर को बनाता है और बनाता है।

इसमें उस class की .class फ़ाइल। (.डॉट) का अर्थ है। फ़ोल्डर उसी directory पर बनाया जाता है जहाँ जावा फ़ाइल save की जाती है या उपयोगकर्ता द्वारा बनाई गई directory पर यह फ़ोल्डर भी बनाया जाता है।

Output :

```
C:\Program Files\Java\jdk1.8.0_111\bin>javac -d .
A.java //Compile class A
```

```
C:\Program Files\Java\jdk1.8.0_111\bin>javac -d .
B.java //Compile class B
```

```
C:\Program Files\Java\jdk1.8.0_111\bin>java
pack2.B //Run class B
class A
```

जावा पैकेज का उपयोग करने के कई फायदे हैं, उनमें से कुछ इस प्रकार हैं -।

classes and interfaces की आसान खोज या पता लगाना।

naming conflicts से बचें। उदाहरण के लिए, दो पैकेजों में छात्र के नाम के साथ दो कक्षाएं हो सकती हैं,

university csdept. Student and college. Itdept. Student

Implement data encapsulation (or data-hiding).

Provide controlled access: संरक्षित और डिफ़ॉल्ट एक्सेस का उपयोग पैकेज स्तर पर नियंत्रण है।

संरक्षित के रूप में घोषित एक सदस्य एक ही पैकेज और उसके उपवर्गों के भीतर कक्षाओं (classes) द्वारा सुलभ (accessible) है।

बिना किसी एक्सेस स्पेसिफायर के एक सदस्य जो डिफ़ॉल्ट स्पेसिफायर है, केवल उसी पैकेज में कक्षाओं द्वारा सुलभ है।

अन्य programs के पैकेज में निहित classes का पुनः उपयोग करें।

अन्य पैकेजों में कक्षाओं (classes) की विशिष्ट तुलना करें।

जावा में एक्सेस स्पेसिफायर के बारे में जानने के लिए आप के बारे में नहीं जानते हैं।

हम समान कक्षाओं को समान पैकेजों में रखते हैं।

उसके बाद, हम केवल import statement के माध्यम से मौजूदा पैकेजों से classes को आयात (import) करते हैं और उन्हें हमारे program में उपयोग करते हैं।

एक पैकेज कुछ classes तक पहुंच प्रदान करता है और अन्य को internal purposes के लिए रखा जाता है।

.Net

.NET का परिचय (Introduction to .NET)

.NET एक सॉफ्टवेयर फ्रेमवर्क है जिसे Microsoft द्वारा विकसित किया गया है। यह एक व्यापक और बहुपरकारी प्लेटफॉर्म है जो विभिन्न प्रकार के एप्लिकेशनों को बनाने और चलाने के लिए उपयोग किया जाता है। .NET का उपयोग वेब एप्लिकेशन, डेस्कटॉप एप्लिकेशन, मोबाइल एप्लिकेशन, गेम्स, क्लाउड-आधारित सेवाओं और अन्य प्रकार के सॉफ्टवेयर विकास के लिए किया जाता है। .NET एक ओपन-सोर्स प्लेटफॉर्म है, जिसे विभिन्न ऑपरेटिंग सिस्टम्स पर चलाने के लिए अनुकूलित किया जा सकता है, जैसे विंडोज, लिनक्स और macOS।

.NET प्लेटफॉर्म को विभिन्न भाषाओं का समर्थन प्राप्त है, जैसे C#, VB.NET, F#, और अन्य, जो डेवलपर्स को एप्लिकेशन डेवलप करने में मदद करती हैं।

.NET के मुख्य घटक (Main Components of .NET)

1. .NET Framework:

.NET Framework एक सॉफ्टवेयर विकास प्लेटफॉर्म है जो मुख्य रूप से Windows ऑपरेटिंग सिस्टम पर आधारित है। इसमें बुनियादी लाइब्रेरी और रनटाइम (CLR - Common Language Runtime) होता है जो एप्लिकेशन को रन करने के लिए आवश्यक संसाधन प्रदान करता है।

2. .NET Core:

.NET Core एक क्रॉस-प्लेटफॉर्म (Windows, macOS, Linux) ओपन-सोर्स संस्करण है। यह .NET Framework का हल्का और अधिक मॉड्यूलर संस्करण है।

3. ASP.NET:

ASP.NET वेब एप्लिकेशन और वेब सेवाओं के निर्माण के लिए एक ढांचा है। इसे विशेष रूप से वेब साइट्स, वेब एप्लिकेशन और API के निर्माण के लिए उपयोग किया जाता है।

4. Xamarin:

Xamarin एक प्लेटफॉर्म है जो डेवलपर्स को C# का उपयोग करके एंड्रॉइड, iOS और Windows मोबाइल एप्लिकेशन बनाने की अनुमति देता है।

5. Mono:

Mono एक ओपन-सोर्स प्लेटफॉर्म है जो लिनक्स और अन्य ऑपरेटिंग सिस्टम्स पर .NET अनुप्रयोगों को चलाने की अनुमति देता है।

6. .NET Standard:

.NET Standard, एक सामान्य API सेट है जिसे .NET Core, .NET Framework, और Xamarin जैसी विभिन्न .NET प्लेटफॉर्म पर साझा किया जाता है। यह विभिन्न .NET प्लेटफॉर्म के बीच कोड साझेदारी को सरल बनाता है।

.NET का इतिहास (History of .NET)

1. .NET की शुरुआत (The Beginning of .NET):

.NET का विकास Microsoft ने 1990 के दशक के अंत में किया था। इसके उद्देश्य थे एक ऐसा प्लेटफॉर्म प्रदान करना जो Windows पर एप्लिकेशन डेवलपमेंट को सरल, प्रभावी और बहुपरकारी बनाए। .NET Framework की पहली बार घोषणा 1999 में की गई थी और इसे 2002 में आधिकारिक रूप से रिलीज़ किया गया था।

2. .NET Framework 1.0 (2002):

.NET Framework 1.0 का पहला संस्करण 2002 में जारी किया गया। इसमें मुख्यतः C#, ASP.NET, और CLR जैसे प्रमुख घटक थे। इसका उद्देश्य विकास को सरल बनाना था, ताकि डेवलपर्स को एप्लिकेशन बनाने में मदद मिल सके।

3. .NET Framework 2.0 (2005):

इस संस्करण में कई सुधार और नई सुविधाएँ जोड़ी गईं। इसमें Generics, ASP.NET 2.0, और Windows Forms के लिए नए फीचर्स शामिल थे। इसके साथ ही CLR की कार्यक्षमता में भी वृद्धि की गई।

4. .NET Framework 3.0 (2006):

इस संस्करण में Windows Communication Foundation (WCF), Windows Presentation Foundation (WPF) और Windows Workflow Foundation (WF) जैसी नई तकनीकों को जोड़ा गया। यह संस्करण अधिक ग्राफिकल और नेटवर्किंग क्षमताओं के साथ आया।

5. .NET Framework 4.0 (2010):

.NET Framework 4.0 में कई सुधार किए गए, जैसे कि Parallel Programming, ASP.NET improvements, और Entity Framework। साथ ही, इसमें dynamic language runtime (DLR) और Improved garbage collection जैसी तकनीकें भी जोड़ी गईं।

6. .NET Core की शुरुआत (2016):

2016 में .NET Core की शुरुआत की गई, जो एक हल्का, मॉड्यूलर और क्रॉस-प्लेटफॉर्म संस्करण था। इसका उद्देश्य डेवलपर्स को एक ऐसा प्लेटफॉर्म देना था जिसे विंडोज़, लिनक्स और मैकओएस पर चलाया जा सके। यह पूरी तरह से ओपन-सोर्स था, जो .NET Framework से अलग था।

7. .NET 5.0 (2020):

.NET 5.0 में .NET Framework और .NET Core का एकीकरण किया गया। इस संस्करण ने .NET के भविष्य के विकास के लिए एक नई दिशा दी। यह एकीकृत प्लेटफॉर्म के रूप में काम करता है, जिसमें वेब, मोबाइल, डेस्कटॉप और अन्य एप्लिकेशन के लिए समर्थन है।

8. .NET 6.0 और आगे (2021 - Present):

.NET 6.0, 2021 में रिलीज़ हुआ और इसमें कई महत्वपूर्ण फीचर्स जैसे कि MAUI (Multi-platform App UI) और C# 10 के लिए समर्थन दिया गया। भविष्य में, .NET का उद्देश्य एक ही कोडबेस से विभिन्न प्लेटफॉर्म पर एप्लिकेशन डेवलप करना है।

.NET के लाभ (Advantages of .NET)

1. क्रॉस-प्लेटफॉर्म सपोर्ट:

.NET Core के माध्यम से, डेवलपर्स अब एक ही कोड बेस से विंडोज़, लिनक्स, और मैकओएस पर एप्लिकेशन चला सकते हैं।

2. आधुनिक विकास उपकरण:

.NET में Visual Studio जैसे शक्तिशाली IDE का समर्थन है, जो डेवलपर्स को बेहतर डेवलपमेंट अनुभव प्रदान करता है।

3. ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग (OOP):

.NET ओब्जेक्ट ओरिएंटेड प्रोग्रामिंग को सपोर्ट करता है, जो सॉफ्टवेयर को अधिक मॉड्यूलर, पुनः उपयोग योग्य और रखरखाव में आसान बनाता है।

4. सुरक्षा:

.NET प्लेटफॉर्म मजबूत सुरक्षा फीचर्स प्रदान करता है, जो एप्लिकेशन को विभिन्न प्रकार के सुरक्षा खतरों से बचाते हैं।

5. समर्थन और समुदाय:

.NET एक व्यापक और सक्रिय समुदाय के साथ आता है, जिससे डेवलपर्स को प्रॉब्लम सॉल्विंग और मदद मिलती है।

निष्कर्ष (Conclusion)

.NET एक शक्तिशाली और बहुपरकारी प्लेटफॉर्म है जो विभिन्न प्रकार के एप्लिकेशन बनाने के लिए उपयोग किया जाता है। इसकी विशेषताएँ, जैसे क्रॉस-प्लेटफॉर्म समर्थन, सुरक्षा, और आधुनिक विकास उपकरण, इसे एक लोकप्रिय विकल्प बनाती हैं। समय के साथ, .NET ने खुद को एक सशक्त और विकसित मंच के रूप में स्थापित किया है, और यह आज भी सॉफ्टवेयर विकास की दुनिया में अपनी जगह बना रहा है।

.NET Framework का उपयोग विभिन्न प्रकार के एप्लिकेशनों को विकसित करने के लिए किया जा सकता है। यह एक व्यापक और बहुपरकारी प्लेटफॉर्म है जो डेवलपर्स को अलग-अलग तरह के एप्लिकेशन बनाने की सुविधा प्रदान करता है। निम्नलिखित कुछ प्रमुख प्रकार के एप्लिकेशन्स हैं, जिन्हें आप .NET Framework का उपयोग करके विकसित कर सकते हैं:

1. डेस्कटॉप एप्लिकेशन (Desktop Applications)

• Windows Forms Application:

यह एप्लिकेशन GUI (Graphical User Interface) आधारित होते हैं। Windows Forms का उपयोग डेस्कटॉप एप्लिकेशनों को बनाने के लिए किया जाता है, जैसे कि सॉफ्टवेयर टूल्स, डेटा प्रोसेसिंग एप्लिकेशन आदि।

• उदाहरण: कैलकुलेटर, डेटा एंट्रिंग सॉफ्टवेयर, आदि।

• WPF (Windows Presentation Foundation):

यह डेस्कटॉप एप्लिकेशनों के लिए एक और शक्तिशाली

प्लेटफॉर्म हैं जो समृद्ध यूजर इंटरफेस (Rich User Interface) प्रदान करता है।

- उदाहरण: ग्राफिकल इंटरफेस वाली एप्लिकेशनों जैसे कि विजुअलाइजेशन टूल्स, इमेज एडिटिंग सॉफ्टवेयर आदि।

2. वेब एप्लिकेशन (Web Applications)

• ASP.NET Web Forms:

ASP.NET Web Forms का उपयोग वेब एप्लिकेशनों को जल्दी और आसानी से बनाने के लिए किया जाता है। इसमें HTML और .NET के घटकों का संयोजन किया जाता है।

- उदाहरण: इंटरनेट पोर्टल्स, ई-कॉमर्स साइट्स आदि।

• ASP.NET MVC (Model-View-Controller):

यह वेब एप्लिकेशन विकास के लिए एक संरचित तरीके से काम करता है, जिसमें डेटा, यूजर इंटरफेस और नियंत्रण को अलग-अलग लेयर में विभाजित किया जाता है।

- उदाहरण: सोशल मीडिया एप्लिकेशन, ब्लॉग प्लेटफॉर्म आदि।

• ASP.NET Core:

यह एक ओपन-सोर्स और क्रॉस-प्लेटफॉर्म फ्रेमवर्क है, जिसे आप विंडोज, लिनक्स और macOS पर वेब एप्लिकेशन और APIs बनाने के लिए उपयोग कर सकते हैं।

- उदाहरण: RESTful APIs, वेब सर्विसेज, क्लाउड-आधारित वेब एप्लिकेशन आदि।

3. मोबाइल एप्लिकेशन (Mobile Applications)

Xamarin:

Xamarin का उपयोग क्रॉस-प्लेटफॉर्म मोबाइल एप्लिकेशन बनाने के लिए किया जाता है। इसमें आप C# का उपयोग करके Android, iOS और Windows फोन के लिए एप्लिकेशन बना सकते हैं।

- उदाहरण: मोबाइल गेम्स, ई-कॉमर्स मोबाइल ऐप्स, हेल्थ ट्रैकिंग ऐप्स आदि।

4. गेम्स (Games)

• Unity:

Unity गेम इंजन में C# का उपयोग किया जाता है और यह .NET Framework के साथ संगत है। आप गेम्स को डेवलप करने के लिए .NET का उपयोग कर सकते हैं, खासकर 2D और 3D गेम्स के लिए।

- उदाहरण: वीडियो गेम्स, VR/AR गेम्स, शैक्षिक गेम्स आदि।

5. वेब सेवाएँ (Web Services)

• ASP.NET Web API:

.NET Framework का उपयोग RESTful Web Services और APIs विकसित करने के लिए किया जा सकता है। इसका उपयोग अन्य एप्लिकेशनों को डेटा प्रदान करने के लिए किया जाता है।

- उदाहरण: मोबाइल ऐप्स के लिए APIs, क्लाउड-सर्वर एप्लिकेशन, आदि।

6. क्लाउड-आधारित एप्लिकेशन (Cloud-based Applications)

- .NET का उपयोग Microsoft Azure के साथ मिलकर क्लाउड-आधारित एप्लिकेशन बनाने के लिए किया जा सकता है। आप क्लाउड प्लेटफॉर्म पर स्केलेबल और सुरक्षित एप्लिकेशन और सेवाएँ विकसित कर सकते हैं।

- उदाहरण: क्लाउड-होस्टेड डेटा एप्लिकेशन्स, SaaS (Software as a Service) एप्लिकेशन, आदि।

7. सिस्टम सॉफ्टवेयर (System Software)

- .NET Framework का उपयोग उन एप्लिकेशनों के लिए भी किया जा सकता है जो हार्डवेयर के साथ इंटरफेस करते हैं और सिस्टम के ऑपरेशन को नियंत्रित करते हैं।

- उदाहरण: नेटवर्क मॉनिटरिंग सॉफ्टवेयर, ऑपरेटिंग सिस्टम टूल्स आदि।

8. डेटाबेस एप्लिकेशन (Database Applications)

• Entity Framework:

यह ORM (Object-Relational Mapping) टूल है जिसका उपयोग डेटाबेस से डेटा को लोड और मैनिपुलेट करने के लिए किया जाता है। इसके साथ आप SQL डेटाबेस के लिए एप्लिकेशनों को विकसित कर सकते हैं।

- उदाहरण: डेटा एनालिटिक्स सॉफ्टवेयर, डेटाबेस प्रबंधन एप्लिकेशन आदि।

9. सुरक्षा एप्लिकेशन (Security Applications)

- .NET Framework का उपयोग सुरक्षा एप्लिकेशनों जैसे कि एंटीवायरस सॉफ्टवेयर, डेटा एन्क्रिप्शन टूल्स, और अन्य सुरक्षा टूल्स को बनाने के लिए किया जा सकता है।

- उदाहरण: डेटा सुरक्षा टूल्स, पासवर्ड मैनेजमेंट एप्लिकेशन, आदि।

.NET प्लेटफॉर्म आर्किटेक्चर :-

एक संरचित ढांचा है जो विभिन्न प्रकार के एप्लिकेशन और सेवाओं के निर्माण के लिए कार्य करता है। यह प्लेटफॉर्म डेवलपर्स को एक प्रभावी, सुरक्षित, और सुसंगत वातावरण प्रदान करता है, जिससे विभिन्न एप्लिकेशन जैसे वेब, डेस्कटॉप, मोबाइल, गेम्स आदि बनाए जा सकते हैं। .NET का आर्किटेक्चर मॉड्यूलर है और विभिन्न घटकों से मिलकर बनता है, जो विकास प्रक्रिया को सरल बनाते हैं।

करता है। यह हल्का और मॉड्यूलर होता है, जिससे यह क्लाउड-आधारित, वेब, और माइक्रो-सर्विसेज जैसे एप्लिकेशन निर्माण में उपयोगी है।

- उदाहरण: ASP.NET Core, .NET Core Web APIs, और अन्य एप्लिकेशन जो विभिन्न प्लेटफॉर्म पर चल सकते हैं।

3. Xamarin

- विवरण: Xamarin का उपयोग क्रॉस-प्लेटफॉर्म मोबाइल एप्लिकेशन बनाने के लिए किया जाता है। यह प्लेटफॉर्म आपको C# का उपयोग करके Android, iOS और Windows मोबाइल एप्लिकेशनों का निर्माण करने की अनुमति देता है।
- उदाहरण: मोबाइल गेम्स, ई-कॉमर्स ऐप्स, और अन्य मोबाइल एप्लिकेशन जो Android और iOS पर काम करते हैं।

4. Entity Framework

- विवरण: Entity Framework एक ORM (Object-Relational Mapping) टूल है जो C# जैसे ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग भाषाओं के डेटा को डेटाबेस के साथ जोड़ता है। इसका उपयोग डेटाबेस से डेटा को क्वेरी और मैनिपुलेट करने के लिए किया जाता है।
- उदाहरण: डेटा-ड्रिवन एप्लिकेशंस जैसे कि अकाउंटिंग सॉफ्टवेयर, ई-कॉमर्स साइट्स आदि।

5. Visual Studio

- विवरण: Visual Studio एक पूर्ण फीचर्ड IDE (Integrated Development Environment) है, जो .NET प्लेटफॉर्म पर एप्लिकेशनों के निर्माण, परीक्षण और डिबगिंग के लिए उपयोग किया जाता है।
- उदाहरण: Windows, Web, और Mobile एप्लिकेशनों के लिए विकास, साथ ही साथ गेम डेवलपमेंट।

6. Visual Studio Code

- विवरण: Visual Studio Code एक हल्का, ओपन-सोर्स कोड एडिटर है जो .NET Core एप्लिकेशनों के लिए उपयुक्त है और विभिन्न भाषाओं और प्लेटफॉर्म पर काम करता है।
- उदाहरण: ASP.NET Core, Node.js, और अन्य एप्लिकेशनों के लिए विकास।

7. Blazor

- विवरण: Blazor .NET का एक वेब फ्रेमवर्क है जो आपको C# का उपयोग करके वेब एप्लिकेशन और इंटरएक्टिव यूजर इंटरफेस (UI) बनाने की अनुमति देता है। यह WebAssembly और .NET Server-Side दोनों मोड्स में काम करता है।
- उदाहरण: रिच वेब एप्लिकेशन जैसे कि इंटरएक्टिव डैशबोर्ड्स, डेटा-ड्रिवन वेब ऐप्स आदि।

.NET Services (.NET सेवाएँ)

1. Azure Services

- विवरण: Microsoft Azure एक क्लाउड कंप्यूटिंग प्लेटफॉर्म है जो .NET डेवलपर्स को .NET एप्लिकेशनों के लिए

क्लाउड सेवाएँ प्रदान करता है। Azure पर आप स्केलेबल वेब एप्लिकेशंस, APIs, डेटा स्टोर और माइक्रोसर्विसेज बना सकते हैं।

- उदाहरण: Azure App Services, Azure Functions, Azure SQL Database, और Azure Kubernetes Service (AKS)।

2. Azure DevOps

- विवरण: Azure DevOps एक सेवा है जो सॉफ्टवेयर विकास की प्रक्रिया को सरल और तेज़ बनाती है। इसमें Continuous Integration (CI) और Continuous Delivery (CD) के टूल्स होते हैं जो .NET एप्लिकेशनों के निर्माण और डिप्लॉयमेंट में मदद करते हैं।
- उदाहरण: पाइपलाइन सेटअप, Git रिपॉजिटरी मैनेजमेंट, टेस्टिंग और डिप्लॉयमेंट।

3. Microsoft Graph

- विवरण: Microsoft Graph API एक सेवा है जो Microsoft 365 डेटा को इंटरएक्ट करने के लिए API प्रदान करती है। इसे .NET एप्लिकेशनों में Microsoft सेवाओं जैसे Outlook, OneDrive, और Teams के साथ इंटीग्रेट किया जा सकता है।
- उदाहरण: Microsoft 365 एप्लिकेशनों के साथ इंटीग्रेटेड एप्लिकेशंस, टीम कॉलिंग, ईमेल और शेड्यूलिंग के लिए एप्लिकेशंस।

4. .NET Microservices

- विवरण: .NET प्लेटफॉर्म का उपयोग माइक्रोसर्विसेज आर्किटेक्चर के निर्माण के लिए किया जा सकता है। माइक्रोसर्विसेज को RESTful APIs, गेटवे और कंटेनर आधारित तंत्र का उपयोग करके डिप्लॉय किया जाता है।
- उदाहरण: स्केलेबल वेब एप्लिकेशंस, मल्टी-चैनल सेवा एप्लिकेशंस आदि।

5. Azure Functions

- विवरण: Azure Functions एक serverless सेवा है जो आपको छोटे, इवेंट-ड्रिवन कोड बनाने की अनुमति देती है, जिसे बिना किसी सर्वर के प्रबंधित किया जाता है। इसे .NET Core का उपयोग करके विकसित किया जा सकता है।
- उदाहरण: इवेंट-ड्रिवन एप्लिकेशन, वेब हुक्स, और API इंटीग्रेशन।

6. Power BI Integration

- विवरण: Power BI Microsoft का एक टूल है जो डेटा विश्लेषण और विज़ुअलाइजेशन के लिए काम आता है। इसे .NET एप्लिकेशनों के साथ जोड़ने के लिए Power BI REST API का उपयोग किया जाता है।
- उदाहरण: डेटा रिपोर्टिंग और डैशबोर्ड्स, बिज़नेस इंटेలిजेंस एप्लिकेशंस।

.net data type:-

.NET डेटा टाइप्स (Data Types in .NET) -

Dear Aspirants, here are the our results in differents exams

(Proof Video Link) 

RAS PRE. 2021 - <https://shorturl.at/qBJ18> (74 प्रश्न , 150 में से)

RAS Pre 2023 - <https://shorturl.at/tGHRT> (96 प्रश्न , 150 में से)

UP Police Constable 2024 - <http://surl.li/rbfyn> (98 प्रश्न , 150 में से)

Rajasthan CET Gradu. Level - <https://youtu.be/gPqDNlc6UR0>

Rajasthan CET 12th Level - <https://youtu.be/oCa-CoTFu4A>

RPSC EO / RO - <https://youtu.be/b9PKjl4nSxE>

VDO PRE. - <https://www.youtube.com/watch?v=gXdAk856Wl8&t=202s>

Patwari - <https://www.youtube.com/watch?v=X6mKGdtXyu4&t=2s>

PTI 3rd grade - https://www.youtube.com/watch?v=iA_MemKKgEk&t=5s

SSC GD - 2021 - <https://youtu.be/2gz2fJyt6vI>

EXAM (परीक्षा)	DATE	हमारे नोट्स में से आये हुए प्रश्नों की संख्या
MPPSC Prelims 2023	17 दिसम्बर	63 प्रश्न (100 में से)
RAS PRE. 2021	27 अक्तूबर	74 प्रश्न आये
RAS Mains 2021	October 2021	52% प्रश्न आये

whatsapp <https://wa.link/f64bo7> 1 web.- <https://rb.gy/86fb4c>

RAS Pre. 2023	01 अक्टूबर 2023	96 प्रश्न (150 में से)
SSC GD 2021	16 नवम्बर	68 (100 में से)
SSC GD 2021	08 दिसम्बर	67 (100 में से)
RPSC EO/RO	14 मई (1st Shift)	95 (120 में से)
राजस्थान S.I. 2021	14 सितम्बर	119 (200 में से)
राजस्थान S.I. 2021	15 सितम्बर	126 (200 में से)
RAJASTHAN PATWARI 2021	23 अक्टूबर (1st शिफ्ट)	79 (150 में से)
RAJASTHAN PATWARI 2021	23 अक्टूबर (2 nd शिफ्ट)	103 (150 में से)
RAJASTHAN PATWARI 2021	24 अक्टूबर (2 nd शिफ्ट)	91 (150 में से)
RAJASTHAN VDO 2021	27 दिसम्बर (1 st शिफ्ट)	59 (100 में से)
RAJASTHAN VDO 2021	27 दिसम्बर (2 nd शिफ्ट)	61 (100 में से)
RAJASTHAN VDO 2021	28 दिसम्बर (2 nd शिफ्ट)	57 (100 में से)
U.P. SI 2021	14 नवम्बर 2021 1 st शिफ्ट	91 (160 में से)
U.P. SI 2021	21 नवम्बर 2021 (1 st शिफ्ट)	89 (160 में से)
Raj. CET Graduation level	07 January 2023 (1 st शिफ्ट)	96 (150 में से)
Raj. CET 12th level	04 February 2023 (1 st शिफ्ट)	98 (150 में से)
UP Police Constable	17 February 2024 (1 st शिफ्ट)	98 (150 में से)

& Many More Exams like UPSC, SSC, Bank Etc.

whatsapp <https://wa.link/f64bo7> 2 web.- <https://rb.gy/86fb4c>