



बेसिक

कम्प्यूटर अनुदेशक

राजस्थान कर्मचारी चयन बोर्ड, जयपुर

भाग - 4

कम्प्यूटर अध्ययन -1

COMPUTER INSTRUCTOR

कम्प्यूटर अध्ययन - 1

S.No.	Chapter Name	Page No.
1.	Fundamental of Computer:	
	• Overview of the computer system	1
	• Types of computer	5
	• Generation of computer	7
	• Input Output devices	10
	• Representation of Data & Number System	25
	• Data Processing	33
• Concepts of files and its types	35	
2.	Data Processing	
	• Word Processing (MS Word)	43
	• Spread Sheet Software (MS Excel)	53
	• Presentation Software (MS Power Point)	65
• DBMS Software (MS Access)	75	
3.	Programming Fundamental	
	• Introduction to C	96
	• Introduction to C++	155
	• Introduction to JAVA	216
	• Introduction to Dot Net	273
	• Artificial Intelligence and Machine Learning	290
	• Python	298
	• Block Chain	318
• Principles and programming Techniques	328	
• Integrated Development Environment and its advantages	335	

Programming Fundamental

Introduction to C language

C language एक programming language है। जिसका use कर के applications बना सकते हैं। इसे unix operating system को दुबारा लिखने के लिए develop किया गया था। क्योंकि unix operating system को B language में लिखा गया था। unix operating system में ज्यादातर program भी B language में लिखे गये थे।

B language को Ken Thompson के द्वारा 1970 में Bell laboratories में develop की गई थी। लेकिन B language ज्यादा popular नहीं हो पायी थी। फिर C language को Dennis Ritchie ने 1972 and 1973 में develop किया था और C language को भी bell laboratories में ही develop किया गया था।

C language एक simple language है। C language के द्वारा ज्यादातर mathematical programs लिखे जाते हैं और C language को popular होने में ज्यादा time नहीं लगा।

C language को Mother language भी कहा जाता है क्योंकि C language के बाद जो भी language बनाई गयी for example (Java, PHP, c#, या C++) इन सभी language में C language का concept use किया गया है।

Versions of C Language

C language को Brain Kernighan and Dennis Ritchie ने 1978 में publish किया था। C language के अब तक कई version आ चुके हैं जो कि निम्न हैं-

- **K & R** - यह C language का original language है। इस version को 1978 में लाया गया था और इस version में standard I/O library जैसी function available थी।
- **ANSI C and ISO C** - इस version को American National Standards institute (ANSI) and International Organization for Standardization (ISO) के द्वारा 1989/1990 में publish किया गया था।
- **C99** - इस version को 1999 में publish किया गया था और इस version में कुछ नये feature add किये गये थे जैसे- inline function, several new data types, long int आदि।
- **C11** - इस version को 2011 में publish किया गया था और इस version में कुछ नये feature add किये गये थे जैसे- library, including type generic macros, anonymous structures आदि।
- **C18** - इस version को June 2018 में publish किया गया था।

Advantage of C language

1. C language एक simple और easy language है। जिसका use आसानी से किया जा सकता है। और C language की सबसे बड़ी खासियत यह है की C language में लिखा गया code बहुत ही fast होता है। यानी C language की execution time fast होती है।
2. C एक structured programming language है और C language में हम functions बना सकते हैं और अपने code को और भी अच्छे से manage कर सकते हैं।
3. C language में C 32 reserved के द्वारा कुछ keyword provide की गयी है। जो ऐसे शब्द हैं, जिनका उपयोग किसी अन्य उद्देश्य के लिए नहीं किया जा सकता है। जिसके लिए वे पूर्वनिर्धारित हैं।
4. C एक middle level language है। जो high level और low level की application बनाने में सक्षम है। यह feature उन programmers के लिए एक advantage है, जो high level और low level की applications create करना चाहते हैं।
5. Assembly language के बाद सबसे fast language C language को ही माना जाता है। इसलिए ये दूसरी programming languages से fast होती है। C language में create की गयी applications की processing बहुत ही fast होती है।

Disadvantage of C language

C language एक powerful language है। लेकिन C language में कुछ कमियाँ भी हैं, जो C language को सीमित बनाती हैं।

1. C language में run time checking नहीं होती है। C language में किसी भी variable के type को identify करने में सक्षम नहीं होती है।
2. C language में re-usability (inheritance) support नहीं करती है और C language में exceptions को run time में handle नहीं किया जा सकता है।
3. C language object oriented programming को support नहीं करता है। जैसे - classes, objects, interfaces आदि।

Usage of C Language

C language का use system application या network drivers बनाने के लिए किया जाता है। क्योंकि यह कोड का उत्पादन करता है। जो Assembly language में लिखे गए कोड के रूप में तेजी से चलता है। C language में उपयोग के लिए कुछ उदाहरण निम्न हैं-

- Operating Systems

- Language Compilers
- Assemblers
- Text Editors
- Print Spoilers
- Network Drivers
- Modern Programs
- Data Bases
- Language interpreters
- Utilities

ऊपर दिये गये उदाहरणों में आप देख सकते हैं कि C language का use किन-किन क्षेत्र में किया जाता है ।

Introduction of Compiler

जब भी हम C language में या C++ में coding करते । तो हमें compiler की जरूरत पड़ती है क्योंकि बिना compiler के code execute नहीं होता है ।

अब चाहे वह code C language में लिखा गया हो या C++ में लिखा गया हो, हमें दोनों language में compiler को use करते हैं ।

C++ C language का advanced version है ।

जब भी हम C language में या C++ में code लिखते हैं, तो हमें code को compile करने के लिए editor की जरूरत पड़ती है । जिसे हम इंटरनेट से download कर सकते हैं ।

C language के code को compile करने के लिए कुछ compiler हम नीचे दे रहे हैं, इन editor को आप इंटरनेट से download कर सकते हैं और अपने PC या laptop में install कर सकते हैं ।

- Dev C++
- Code Blocks
- Turbo C

ऊपर दिये गये उदाहरणों में तीन editor provide किये हैं, जिसका use coding को compile करने के लिए किया जाता है ।

Dev C++ compiler

इस editor के द्वारा आप C language या C++ के code को बड़े ही आसानी से compile कर सकते हैं और यह editor Dev C++ का एक Integrated Development Environment है, जो

graphical user interface provide करती है और साथ ही साथ syntax highlighting और automatic code completion की facilities भी provide करती है ।

Code Blocks Compiler

Code Blocks भी एक editor है, जो C language या C++ के code को compile करने का काम करता है । इस editor में भी हमें IDE provide की जाती है और साथ ही साथ syntax highlighting और automatic code completion की facilities भी provide की जाती है और इस editor को ज्यादातर programmer ही use करते हैं ।

Turbo C Compiler

Turbo c compiler भी एक editor है, जो C language या C++ के code को compile करने का काम करता है । लेकिन इस editor में हमें syntax highlighting और automatic code completion की facilities नहीं मिलती है । इस editor में हमें खुद code को type करना पड़ता है और इस editor को ज्यादातर school और collages में use किया जाता है ।

Working with Compilers

- जब भी हम Compiler install करते हैं तो हमें code लिखने के लिए project बनाना पड़ता है और जब हम उस project को save करते हैं तो उस project का type यानि file का extension C या CPP में save करते हैं नही तो हमारा code work नहीं करता है ।
- जब हम C language में code करना चाहते हैं तो हम file का extension type C provide करते हैं और जब हम C++ में code करना चाहते हैं तो extension type CPP provide करते हैं ।
- Dev C++ editor या code blocks editor में हमें project create करने की facility मिलती है और project create करने के बाद C language या C++ का general syntax खुद ही include हो जाता है ।
- Dev C++ editor या code blocks editor में empty file भी create करने की facility provide की गयी है लेकिन empty file create करने के बाद C language या C++ के basic general code को हमें खुद ही type करना पड़ता है ।
- Turbo C editor में हमें project create करने की facility नहीं मिलती है, तो इस editor में हमें empty file create करना पड़ता है और खुद ही code type करना पड़ता है ।

Introduction of Tokens

अगर आप C language में coding करना चाहते हैं । तो C language का syntax समझना आपके लिए बहुत आवश्यक है । एक C program tokens से मिलकर बना होता है । यदि आप C language के सभी tokens को use करना सीख जाते हैं, तो आप C language के code को बहुत ही fast type कर सकते हैं ।

उदाहरण के लिए नीचे एक programs दिया जा रहा है इस programs में tokens को सही क्रम में कैसे लिखते हैं, उसे दिखा रहे हैं ।

```
#include <stdio.h>

#include <conio.h>

Int main()
{
    printf(“programing language is the best”);
}
```

अगर आप ऊपर दिये गये codes को देखें तो int, main, printf एक tokens ही है, जो सही क्रम में लिखे गये हैं और Tokens कुल 6 प्रकार के होते हैं । इनकी list हम नीचे दे रहे हैं ।

- Identifiers
- Operators
- Variables
- Strings
- Keywords
- Constants

Identifiers

C language में Identifiers वह नाम होते हैं, जिसका use variables, constants और functions में किया जाता है । इन नामों को use करने के लिए कुछ rule follow करना पड़ता है । अगर आप इन rules को follow नहीं करेंगे तो programs में error आ जाते हैं ।

- Identifiers में % @, - characters symbol का use नहीं किया जाता है ।
- C language एक case sensitive language है । इसलिए num और Num यह दो different identifiers नामों से जाने जाते हैं ।
- Identifiers में हम operators का use नहीं कर सकते हैं ।
- Identifiers को Digits से start नहीं किया जाता है । Identifiers की शुरुआत character या underscore से कर सकते हैं ।
- C language में A से लेकर Z तक small और capital दोनों तरह के letters का use कर सकते हैं ।

Example of Identifiers

Identifiers के कुछ example नीचे दिये जा रहे हैं ।

-name //right Identifiers
 2num //wrong Identifiers
 Book-name //wrong Identifiers
 Book-1-name //right Identifiers

Keywords

C language में कुछ keywords को predefined tokens या reserved words कहा जाता है । C language हमें 32 keywords provide करती है, जो हर keyword का use particular task perform करने के लिए define किया गया है । इन Keywords का use किसी अन्य कामों में नहीं किया जा सकता है क्योंकि यह एक predefined tokens या reserved keywords होती है ।

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

Character Set

C language में हम characters को एक sequence में लिखते हैं । जैसे- #include <stdio.h> यह statement characters से मिलकर बना हुआ है । इस statement में i,n,c,l और h एक characters हैं ।

C programming में कोई भी characters use कर सकते हैं लेकिन इसके लिए character set define किया गया है । जो character इस set में आते हैं केवल वही character use किये जा सकते हैं ।

Alphabet

Alphabet characters में हम A से Z capital case और a से z small case characters को use कर सकते हैं ।

Uppercase: A B C X Y Z

Lowercase: a b c x y z

Digits

Digits में आप 0 से लेकर 9 तक कोई भी digit use कर सकते हैं ।

0 1 2 3 4 5 6 7 8 9

Special Characters

Special Characters- नीचे दिए गए symbol characters को use कर सकते हैं ।

,	<	>	.	-
()	;	\$:
%	[]	#	?
'	&	{	}	"
^	!	*	/	
_	\	~	+	@

Introduction of Data Types

जब भी हम C language में coding करते हैं, तो हमें Variables create करने पड़ते हैं और इन्हीं Variable के type को data types कहते हैं ।

- C language हो या C++ हो बिना data type declare किये हम कोई भी प्रोग्राम नहीं बना सकते हैं । अगर हमें किसी भी तरह का प्रोग्राम बनाना है, तो उसके लिए हमें variables create करना पड़ता है और अगर variables create नहीं करते हैं, तो ज्यादा से ज्यादा हम statement print कर सकते हैं ।
- C language में मुख्यतः तीन प्रकार के data types पाए जाते हैं । पहला int (integer) और दूसरा float होता है और तीसरा char होता है । जिसे हम character type data कहते हैं ।
- C language में हम जो भी variables create करते हैं । वो शारे के शारे variables कुछ न कुछ memory consume करते हैं । यानी space लेते हैं ।

Programming Language- C

- C language में create किए जाने वाले variables (data types) का list नीचे दिया जा रहा है । जिनका use अलग अलग कामों के लिए किया जाता है ।
- Integer Types
 - int
 - short int
 - long int
 - signed int
 - unsigned int
- Floating Point types
 - float
 - double
- Character Types
 - char

Integer Data Types

इस प्रकार के data types में only numbers store किये जाते हैं पर ध्यान रहे integer data type में बिना दशमलव के numbers store किये जाते हैं और अगर दशमलव के साथ numbers store करेंगे तो program में error आता है । Integer data types 5 प्रकार के होते हैं और इन्हें memory size और range के अनुसार बाँटा गया है ।

Data type	Size (bytes)	Range
int	2	-32768 से 32767
short int	1	-128 से 127
long int	4	-2,147,483,648 से 2,174,483,647
signed int (negative value के लिए)	2	-32768 से 32767
unsigned int	2	0 से 65535

Floating Point Data Types

Floating point data types में हम number को दशमलव के साथ store कर सकते हैं । Floating point data types दो प्रकार के होते हैं । पहला तो float ही होता है और दूसरे को हम double data types कहते हैं ।

Floating data types में दशमलव के बाद 7 digits तक number store कर सकते हैं और Double data types में दशमलव के बाद 17 digits तक number store कर सकते हैं ।

Data type	Size (bytes)	Range
float	4	3.4E-38 से 3.4E+38
double	7	1.7E-308 से 1.7E+308

Character Data Types

Character Data Types में हम only एक ही Character store कर सकते हैं और जब भी हम किसी Character को store करते हैं तो उसे single quotes (") में type करते हैं। Character data types को दो categories में divide किया गया है।

Data type	Size (bytes)	Range
char	1	128 से 127
unsigned char	1	0 से 255

Void Data Type

Void Data type को ज्यादातर function में use किया जाता है for example - void main() {} यहाँ पर void एक data types है।

अगर आपका function कोई value return नहीं करता है, तो उसका return type void define करते हैं। उदाहरण के लिए आप इस प्रकार से function define कर सकते हैं जैसे- void myFunction():

अगर आप function में कोई parameters पास नहीं करते हैं। तो उसकी जगह पर void define कर सकते हैं। Void type से यह पता चलता है कि इस function में कोई argument नहीं लिया गया है। और इस प्रकार void को parameter के रूप में pass किया जाता है। int myFunction(void);

Declare the variables and function

हम data types और function को कैसे declare करते हैं इसका एक example नीचे दिया गया है।

Example

```

void main()
{
int age = 20;
float year = 2019.0;

```

Programming Language- C

```
double num = 2019.00;
```

```
char name = 'R';
```

```
}
```

Introduction of Variables

- जब भी हम data के साथ कोई operations perform करना चाहते हैं, तो सबसे पहले data को computer के memory में store करना पड़ता है। Computer memory में data को store करने के लिए हमें variables को create करना पड़ता है।
- जब भी हम Variables create करते हैं, तो यही जरूरी नहीं की वही पर value भी assign कर दे अगर हम चाहे तो वही पर value भी assign कर सकते हैं।
- एक variable memory में किसी location का नाम होता है। यह नाम उस memory location को uniquely identify करने के लिए use किया जाता है और इसी नाम के द्वारा उस memory location में data store करते हैं और उस data को पुनः प्राप्त करते हैं।
- आप जिस तरह का variables create करते हैं, उसी प्रकार से memory में space मिलता है। यानी अगर आप int (integer) variables create करते हैं तो 2 bytes variable को memory में compiler allot करेगा।
- इसके बाद हमें उस memory location का एक नाम define करना पड़ता है। ताकि जब भी चाहे उस memory location में store की गई value को उस नाम के द्वारा access कर सके और इसी नाम variable कहा जाता है।
- Variables की values changeable होती है। आप एक value को हटाकर दूसरी value डाल सकते हैं और program execution के दौरान भी कर सकते हैं।

Creating Variables

जब भी हम कोई variables create करते हैं, तो सबसे पहले हमें data types define करना पड़ता है फिर उसके बाद उस data types का नाम देते हैं। अगर हम चाहे तो वही पर value भी assign कर सकते हैं। अब नीचे दिये गये example को देखें।

```
<data-type> <variable-name>;
```

```
<data-type> <variable-name> = <variable value>;
```

```
int age;
```

```
int age = 20;
```

Scope of Variables

Variable को दो categories में divide किया गया है। यानी हम variables को पूरे program में कहाँ declare कर सकते हैं। इसी को हम scope of variables कहते हैं।

Local Variables

Local Variables वो Variables होते हैं, जो program के किसी function में define किये जाते हैं और उन variables का use only उसी function में किया जा सकता है। जिस function में वह variables create किया गया है। इस variables का use किसी दूसरे function में नहीं कर सकते हैं।

Example

```
#include <stdio.h>
#include <conio.h>
void main Function ();
int main()
{
    int num1=10;
    printf(" this is num1 value: %d", num1)
    main Function();
    getch ();
}
void main Function();
{
    /* local variable num2 */
    int num2=15;
    printf(" this is num2 value: %d", num2)
}
```

Programming Language- C

output

this is num1 value : 10

this is num2 value : 15

Global Variables

Global Variables वो Variables होते हैं, जिनका scope पूरे program में होता है। इन variables को आप पूरे program में कहीं भी access कर सकते हैं। इन variables को program की शुरुआत में ही define कर दिया जाता है।

```
#include <stdio.h>

#include <conio.h>

int num3=50; /* num3 is a global variable */

void main Function ();

int main()
{
    int num1=10;
    printf(" this is num1 value: %d", num1)
    main Function();
    printf(" this is num3 value: %d", num3)
    getch ();
}

void main Function();
{
    int num2=15;
    printf(" this is num2 value: %d", num2)
}

output
```

this is num1 value : 10

this is num2 value : 15

this is num3 value : 50

Introduction of Constants

C language में Constants वो variables होते हैं, जिनकी value change नहीं होती है। जब हम कोई constant declare करते हैं। तो उसकी value fixed रहती है। यदि इसकी value change करने की कोशिश की जाती है, तो program में error आ जाता है। यानी constant variables की value fix होती है। जो program के run time में भी इसकी value change नहीं की जा सकती है।

C language में Constant दो प्रकार के होते हैं। पहला Constant Literals और दूसरा Constant Variables इन दोनों constant के बारे में नीचे details दिया जा रहा है।

- Constant Literals
- Constant Variables

C Language - Constant Literals

C language में Constant literals ऐसी values होती हैं जिन्हें हम program में directly use कर सकते हैं। अब नीचे दिए गए code को देखिये।

```
x = y+5;
```

- ऊपर दिए गए code में 5 एक constant literals हैं, इसे program में directly use कर सकते हैं। जब हम program run करेंगे तो इसका value change नहीं होगा।
- Constant literals को हम integer नंबर ही मानते हैं। क्योंकि इसकी value changeable नहीं होती है। इसलिए Constant literals को कहीं-कहीं use किया जाता है।
- उदाहरण के लिए मान लीजिये आपने एक program बनाया और उस program में Constant literals का use किया है। अगर आपको Constant literals की value change करनी हो तो आपको पूरे program में Constant literals को search करना पड़ेगा उसके बाद ही value change कर सकते हैं।

C Language - Constant Variables

Constant variables वो variables होते हैं। जैसे हम खुद declare करते हैं और Constant variables का सबसे बड़ा benefit यह होता है। अगर Constant variables की value change करना हो तो बड़े ही आसानी से कर सकते हैं। Constant variables को हम दो प्रकार से declare कर सकते हैं। इसका details नीचे दिया जा रहा है।

Programming Language - C

- pre processor (#define)
- const Keyword

C Language - Using pre processor (#define)

C Language में #define एक pre processor है। इसका use करके हम constant variables declared कर सकते हैं। #define pre processor के द्वारा constant variable main function के पहले ही create किये जाते हैं। pre processor के द्वारा define किये गए constant variables को program में कहीं भी use कर सकते हैं और जरूरत पड़ने पर इसका value भी change कर सकते हैं।

Example

```
#include <stdio.h>
#include <conio.h>
#define num 20      /* this is constant variable num */
int main()
{
    printf(" your Result is : %d", num);
    return 0;
}
```

output

Your Result is : 20

C Language - Using const Keyword

C Language में constant variables declare करने के लिए हम const keyword का भी use कर सकते हैं। आप जैसे ही constant keyword का use करेंगे आपका constant variables create हो जायेगा और constant variables को हम function में भी create कर सकते हैं और उसे use भी कर सकते हैं। अब नीचे दिये गये code को देखें और समझें।

Example

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



```
int main()  
{  
    const int a = 10;  
    const int b=10;  
    int c;  
    c = a+b;  
    printf(" your Result is : %d", c);  
    return 0;  
}
```

output

Your Result is : 20

Type of C Constant

C Language में 5 प्रकार के constant पाए जाते हैं। जिनकी list नीचे दी गयी है और C Language में escape sequence characters को भी constants ही माना जाता है।

1. Integer Constants
2. Character Constants
3. String Constants
4. Floating-point Constants
5. Enumeration Constants

Introduction of Flowchart

Flow chart हमारे program का graphical representation होता है। जब भी C language में कोई program लिखते हैं, तो बिना flow chart के लिखते हैं। अगर हमें अपना program किसी दूसरे व्यक्ति को समझाना हो तो program समझ नहीं पायेगा। इसलिए हम Flow chart के माध्यम से अपने program को graphically represent करते हैं। ताकि कोई भी program के logic को आसानी से समझ सके।