C ભાષામાં ડેટા પ્રકારો

પરિચય

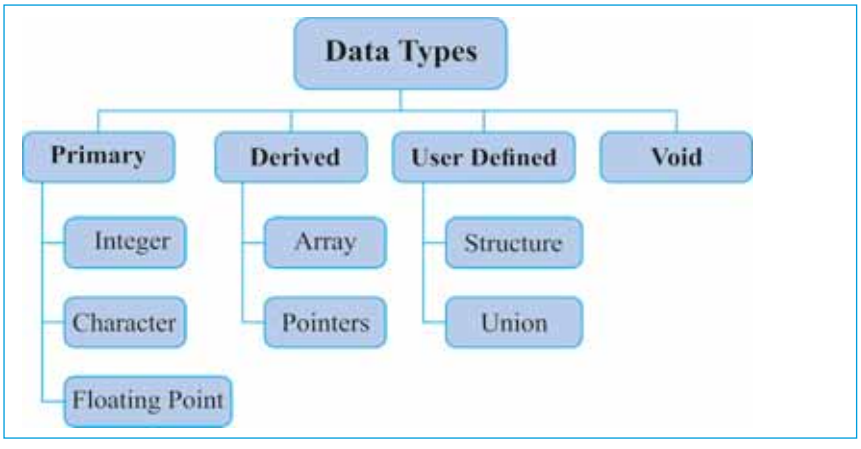
અગાઉના પ્રકરણે C પ્રોગ્રામિંગનો પરિચય આપ્યો હતો. હવે આપણે ડેટા પ્રકારોને સમજીને આપણી C પ્રોગ્રામિંગ યાત્રાને આગળ વધારીએ. C પ્રોગ્રામિંગમાં, ડેટા પ્રકારો ખૂબ જ મહત્વપૂર્ણ છે કારણ કે તે કમ્પ્યૂટરને જણાવે છે કે આપણે કયા પ્રકારનો ડેટા વાપરી રહ્યા છીએ. તે યોગ્ય વસ્તુનો સંગ્રહ કરવા માટે યોગ્ય કન્ટેનર (પાત્ર) વાપરવા જેવું છે - ઉદાહરણ તરીકે, પાણી માટે બોટલ અને ખાંડ માટે બરણી. C પ્રોગ્રામિંગમાં પૂર્ણાંક સંખ્યાઓ, દશાંશ (અપૂર્ણાંક) સંખ્યાઓ અને અક્ષરો જેવા વિવિધ પ્રકારના મૂલ્યોનો સંગ્રહ કરવા માટે જુદા જુદા ડેટા પ્રકારોનો ઉપયોગ થાય છે. દરેક ડેટા પ્રકાર કમ્પ્યૂટરને જણાવે છે કે તેને કેટલી મેમરીની જરૂર છે અને તે ડેટાનો ઉપયોગ કેવી રીતે કરવો. યોગ્ય ડેટા પ્રકારનો ઉપયોગ આપણને સચોટ અને કાર્યક્ષમ પ્રોગ્રામ લખવામાં મદદ કરે છે. આ પ્રકરણમાં C પ્રોગ્રામિંગમાં વિવિધ ડેટા પ્રકારોના ઉપયોગનો પરિચય આપવામાં આવ્યો છે, જેમાં મૂળભૂત (basic), ડિરાઈવ્ડ (derived), યુઝર-નિર્મિત (user-defined) અને ખાલી (empty) (void) ડેટા પ્રકારોનો સમાવેશ થાય છે. આપણે સ્ટોરેજ ક્લાસ (storage class) વિશે પણ ચર્ચા કરીશું, જે મેમરીમાં ચલ કેવી રીતે વર્તે તે નિયંત્રિત કરે છે. આપણે જાણીશું કે ડેટા પ્રકારો શું છે, C માં તેમનો ઉપયોગ કેવી રીતે થાય છે અને આપણા પ્રોગ્રામ માટે યોગ્ય ડેટા પ્રકાર કેવી રીતે પસંદ કરવો.

ડેટા પ્રકારો (Data Types)

C પ્રોગ્રામિંગમાં ડેટા ટાઇપને ચાર મુખ્ય જૂથોમાં વહેંચવામાં આવ્યા છે:

1. **મૂળભૂત ડેટા ટાઇપ (Primary Data Types) :** આ બેઝિક ડેટા ટાઇપ છે, જે સરળ મૂલ્યો રાખવા માટે વપરાય છે જેમ કે પૂર્ણાંક *(int)*, અક્ષર *(char)* અને અપૂર્ણાંક સંખ્યા *(float)*.
2. **ડિરાઈવ્ડ ડેટા ટાઇપ (Derived Data Types) :** આ મૂળભૂત ડેટા ટાઇપ પરથી બનેલા હોય છે અને તેમાં એરે (array) અને પોઈન્ટર (pointer) નો સમાવેશ થાય છે, જે આપણને વધુ જટિલ ડેટાને સંભાળવામાં મદદ કરે છે.
3. **યુઝર-નિર્મિત ડેટા ટાઇપ (User-Defined Data Types):** આ પ્રોગ્રામરને પોતાની જરૂરિયાત મુજબ નવા ડેટા માળખાં (data structures) બનાવવા દે છે, જેમ કે *struct, union* વગેરે.
4. **ખાલી ડેટા ટાઇપ (Void Data Type) :** *void* પ્રકાર જે "કોઈ મૂલ્ય નથી" એવો અર્થ દર્શાવે છે અને સામાન્ય રીતે પરત મૂલ્ય ન આપતા ફંક્શન માટે વપરાય છે.

તદુપરાંત, સ્ટોરેજ ક્લાસ (storage class) જેમ કે ઓટો (*auto*), સ્ટેટીક (*static*), ગ્લોબલ (*global*) અને રજીસ્ટર (*register*) નક્કી કરે છે કે કોઈ ચલ મેમરીમાં કેટલો સમય રહેશે અને તે પ્રોગ્રામના ક્યાં ભાગમાં ઉપલબ્ધ હશે. આકૃતિ 7.1 વિવિધ ડેટા ટાઇપ દર્શાવે છે.

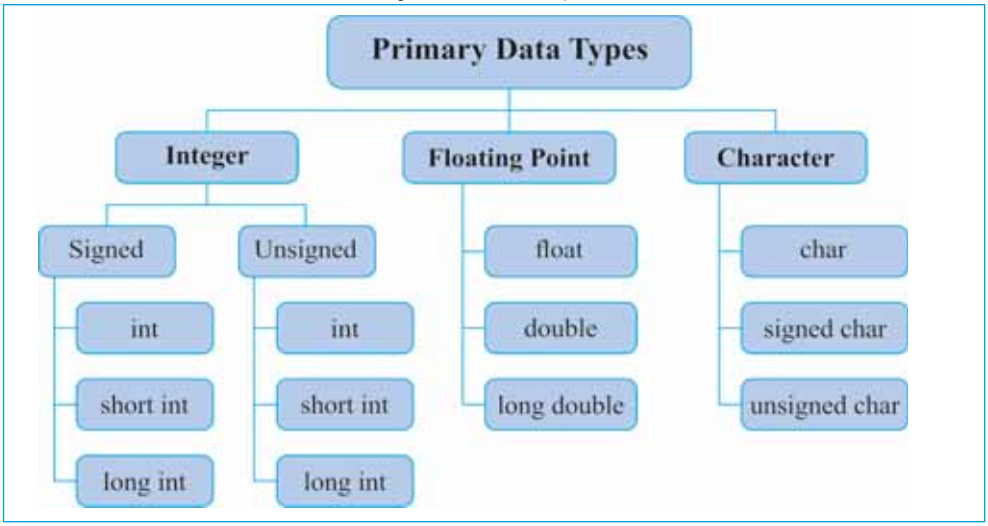


આકૃતિ 7.1 : ડેટા પ્રકારો

Get More Learning Materials Here : CLICK HERE www.studentbro.in

મૂળભૂત ડેટા ટાઈપ (Primary Data Types)

ચાલો, આકૃતિ 7.2 મુજબ મૂળભૂત ડેટા ટાઈપ (Primary Data Types) સમજીએ. મૂળભૂત ડેટા ટાઈપ એ એવા મૂળ પ્રકારના ડેટા છે, જે C પ્રોગ્રામિંગમાં પાયારૂપ છે. જેમ, આપણે અગાઉ ચર્ચા કરી હતી કે C ભાષામાં વિવિધ પ્રકારના ડેટા અલગ-અલગ પ્રકારના મૂલ્યો (સંખ્યા, અક્ષર વગેરે) સંગ્રહવા માટે વપરાય છે. C માં મુખ્ય ચાર પ્રકારના મૂળભૂત ડેટા ટાઈપ છે: *int* (integer) - પૂર્ણ સંખ્યાઓ સંગ્રહવા માટે, *float* (floating-point) - અપૂર્ણાંક (દશાંશ) સંખ્યાઓ સંગ્રહવા માટે, *char* (character) - એક અક્ષર અથવા પ્રતીક (single letter or symbol) સંગ્રહવા માટે, *double* - વધુ ચોકસાઈવાળી અપૂર્ણાંક સંખ્યાઓ (extra-precise decimal values) માટે. પ્રત્યેક પ્રકારનો પોતાનો અલગ ઉપયોગ અને મેમરીની જરૂરિયાત હોય છે, જે ટેબલ 7.1માં દર્શાવવામાં આવી છે.



આકૃતિ 7.2 : મૂળભૂત ડેટા પ્રકારો

ડેટા ટાઈપ (Data Type)	વર્ણન	ઉદાહરણ
int	પૂર્ણાંક સંખ્યાઓ (ધન અથવા ઋણ) સંગ્રહે છે.	int rollNumber = 25;
float	મર્યાદિત ચોકસાઈ સાથે અપૂર્ણાંક (દશાંશ) સંખ્યાઓ સંગ્રહે છે.	float height = 5.6;
double	float કરતાં વધુ ચોકસાઈ સાથે અપૂર્ણાંક સંખ્યાઓ સંગ્રહે છે.	double distance = 123.456789;
char	' ' માં મૂકેલ એક અક્ષર (single character) ને સંગ્રહે છે.	char grade = 'A';

ટેબલ 7.1 : મૂળભૂત ડેટા પ્રકારો

int ડેટા પ્રકારો (int Data Type)

int ડેટા પ્રકાર C પ્રોગ્રામિંગમાં સૌથી મહત્વપૂર્ણ અને વારંવાર વપરાતા ડેટા પ્રકારોમાંનો એક છે. તે પૂર્ણાંક સંખ્યાઓ (whole numbers) - ધન, ઋણ કે શૂન્ય - સંગ્રહવા માટે વપરાય છે. ઉદાહરણ તરીકે, જો આપણે લખીએ: **int age = 15;** તો તેનો અર્થ થાય છે કે કમ્પ્યૂટરને age નામના ચલમાં પૂર્ણાંક 15 સંગ્રહવા કહેવામાં આવ્યું છે. C પ્રોગ્રામિંગમાં *int* ના વિવિધ પ્રકારો ઉપલબ્ધ છે (જે ટેબલ 7.2માં દર્શાવેલ છે), જે નાની કે મોટી સંખ્યાઓને સંભાળી શકે છે. આથી, પ્રોગ્રામરને પોતાની જરૂરિયાત મુજબ યોગ્ય પ્રકાર પસંદ કરવાની લવચીકતા (flexibility) મળે છે. વસ્તુઓની ગણતરી કરવી હોય, સ્કોર ટ્રેક કરવો હોય કે માત્રાઓ (quantities) સંગ્રહવી હોય — તમામ કિસ્સાઓમાં પૂર્ણ સંખ્યાઓ સાથે કામ કરવા માટે *int* ડેટા પ્રકાર C માં સૌથી પ્રાથમિક પસંદગી છે.

વાક્યરચના (Syntax):

```
data-type identifier;
Example : int number = 3;
```


Get More Learning Materials Here : CLICK HERE www.studentbro.in

પ્રકાર (Type)	વર્ણન	ઉદાહરણ
int	મૂળભૂત પૂર્ણાંક પ્રકાર (પૂર્ણ સંખ્યાઓ માટે)	int x = 100;
short int	નાની સાઈઝનો પૂર્ણાંક પ્રકાર (સામાન્ય રીતે 2 બાઈટ)	short int a = 10;
long int	મોટી સાઈઝનો પૂર્ણાંક પ્રકાર (સામાન્ય રીતે 4 બાઈટ)	long int b = 100000;
unsigned int	ફક્ત ધન સંખ્યાઓ માટે (ઋણ મૂલ્યો નહીં)	unsigned int c = 50;
signed int	ધન અને ઋણ બંને પ્રકારની સંખ્યાઓ માટે (મૂળભૂત રીતે)	signed int d = –25;

ટેબલ 7.2 : int ડેટા પ્રકારો

```
/* C program to illustrate sum of two integer numbers.*/
#include <stdio.h>
int main()
{
    int number1 = 25;
    int number2 = 35;
    int sum;
    sum = number1 + number2;
    printf("The sum of %d and %d is %d", number1, number2, sum);
    return 0;
}
Result:
The sum of 25 and 35 is 60
```

ચાલો, આ C પ્રોગ્રામને સમજીએ. C પ્રોગ્રામમાં *main()* ફંક્શન આવશ્યક છે — પ્રોગ્રામનું અમલીકરણ (execution) અહીંથી શરૂ થાય છે. *main()* ના { } કર્લી બ્રેકેટની અંદર, આપણે બે પૂર્ણાંક ચલ number1 અને number2 ઘોષિત કરીએ છીએ અને તેમને અનુક્રમે 25 અને 35 મૂલ્યો આપીએ છીએ. ત્યારબાદ, આપણે એક અન્ય ચલ sum લઈએ છીએ, જેમાં આ બંને સંખ્યાઓનો સરવાળો સંગ્રહ કરાય છે. વાસ્તવિક સરવાળો sum = number1 + number2 લાઈનમાં થાય છે. છેલ્લે, *printf()* ફંક્શનનો ઉપયોગ કરીને પરિણામ બતાવીએ છીએ. અહીં %d ફોર્મેટ સ્પેસિફાયરનો અર્થ છે "પૂર્ણ સંખ્યા દર્શાવો". આ સરળ ઉદાહરણ બતાવે છે કે કેવી રીતે આપણે સંખ્યાઓ સંગ્રહવી, ગણતરી કરવી અને પરિણામ દર્શાવવું - આ બધું C પ્રોગ્રામિંગમાં કરી શકીએ છીએ. હવે ચાલો, unsigned integer સંખ્યાઓ વાપરતો પ્રોગ્રામ લખીએ.

```
/* C program to illustrate use of unsigned int.*/

#include <stdio.h>
int main()
{
    unsigned int a = 300;
    printf("The value of unsigned int a is: %u\n", a);

    return 0;
}
Result:
The value of unsigned int a is: 300
```


Get More Learning Materials Here : CLICK HERE www.studentbro.in

unsigned int ડેટા પ્રકાર ફક્ત ધન પૂર્ણ સંખ્યાઓ (positive whole numbers) જ સંગ્રહ કરે છે તે ઋણ મૂલ્યો (negative values) સંગ્રહ કરી શકતું નથી. આ પ્રોગ્રામમાં, આપણે ચલ a ને મૂલ્ય 300 આપીએ છીએ અને તેને %u ફોર્મેટ સ્પેસિફાયરથી પ્રિન્ટ કરીએ છીએ, જે ખાસ કરીને unsigned integers માટે વપરાય છે. નીચેનો પ્રોગ્રામ signed integers નો ઉપયોગ દર્શાવે છે.

```
/* C program to illustrate use of signed int.*/

#include <stdio.h>
int main()
{
    signed int b = -200;
    printf("The value of signed int b is: %d\n", b);

    return 0;
}
Result:
The value of signed int b is: -200
```

signed int એ C ભાષામાં મૂળભૂત (default) પૂર્ણાંક પ્રકાર છે. તે ધન તેમજ ઋણ બંને પ્રકારની સંખ્યાઓ સંગ્રહી શકે છે. અહીં, આપણે વેરિયેબલ b ને મૂલ્ય -200 આપીએ છીએ અને તેને %d ફોર્મેટ સ્પેસિફાયર વડે પ્રિન્ટ કરીએ છીએ, જે signed integers માટે વપરાય છે.

float અને double ડેટા પ્રકારો (float and double Data Types)

C માં *float* ડેટા પ્રકારનો ઉપયોગ દશાંશ ચિહ્નવાળી સંખ્યાઓ સંગ્રહ કરવા માટે થાય છે, જેમ કે 3.14 અથવા 89.5. તે C ના મૂળભૂત ડેટા પ્રકારમાંનો એક છે અને જ્યારે આપણને માપ, ટકાવારી અથવા દશાંશની ચોકસાઈની જરૂર હોય તેવી કોઈપણ ગણતરીઓ સાથે કામ કરવાની જરૂર હોય ત્યારે તે યોગ્ય છે. ઉદાહરણ તરીકે, જો આપણે કોઈનો ટેસ્ટ એવરેજ 87.65 ને સંગ્રહ કરવા માંગતા હોઈએ, તો આપણે આનો ઉપયોગ કરીશું: **float average = 87.65;.** રોજિંદી મોટાભાગની દશાંશ ગણતરીઓ માટે *float* પ્રકાર યોગ્ય છે, પરંતુ જ્યારે આપણને તેનાથી પણ વધુ ચોકસાઈની જરૂર હોય, ત્યારે C *double* નામનો બીજો પ્રકાર ઓફર કરે છે, જે ટેબલ 7.3 માં દર્શાવેલ છે. આ દશાંશ સંખ્યાના પ્રકારો કોઈપણ પ્રોગ્રામ માટે આવશ્યક છે, જે વાસ્તવિક દુનિયાના મૂલ્યો સાથે કામ કરે છે, જ્યાં ફક્ત પૂર્ણાંક સંખ્યાઓ પૂરતી નથી!

પ્રકાર (Type)	વર્ણન	ઉદાહરણ
float	દશાંશ મૂલ્યોને ~6 અંકોની ચોકસાઈ સાથે સંગ્રહે છે (4 બાઈટ)	float price = 10.75;
double	દશાંશ મૂલ્યોને ~15 અંકોની ચોકસાઈ સાથે સંગ્રહે છે (8 બાઈટ)	double pi = 3.14159265359;
long double	દશાંશ મૂલ્યોને તેનાથી પણ વધુ ચોકસાઈ સાથે સંગ્રહે છે (સામાન્ય રીતે 10 બાઈટ)	long double value = 3.141592653589793238;

ટેબલ 7.3 : float અને double ડેટા પ્રકારો

Get More Learning Materials Here : CLICK HERE www.studentbro.in

ચાલો આપણે તેમને ઉદાહરણ સાથે સમજીએ.

```
/* C program to illustrate multiplication of two float or double
numbers. */
#include <stdio.h>
int main()
{
  float num1 = 5.5;
  double num2 = 4.25;
  double result;
  result = num1 * num2;
  printf("The multiplication of %f and %f is %f", num1, num2,result);
  return 0;
}
Result:
The multiplication of 5.500000 and 4.250000 is 23.375000
```

ચાલો ઉપરના પ્રોગ્રામને સમજીએ, જ્યાં *float* ચલ num1 ઘોષિત કરવામાં આવ્યો છે અને તેને 5.5 ની કિંમત આપવામાં આવી છે. તેવી જ રીતે, એક *double* ચલ num2 ઘોષિત કરવામાં આવ્યો છે અને તેને 4.25 ની કિંમત સાથે આરંભિત કરવામાં આવ્યો છે. ગુણાકારના પરિણામને સંગ્રહ કરવા માટે result નામનો બીજો *double* ચલ ઘોષિત કરવામાં આવ્યો છે. વિધાન (statement) result = num1 * num2; એ *float* અને *double* મૂલ્યોનો ગુણાકાર કરે છે અને પરિણામને result ચલમાં સંગ્રહ કરે છે. છેલ્લે, પ્રોગ્રામ return 0; સાથે સમાપ્ત થાય છે, જે સફળતાપૂર્વક અમલ થયાનો સંકેત આપે છે. ચાલો એક વધુ ઉદાહરણ લઈએ.

```
/* C program to find radius of a circle. */
#include <stdio.h>
#include <math.h>
int main()
{
    double area = 78.5;  double pi = 3.14;
    double radius;
    radius = sqrt(area / pi);
    printf("Radius using double: %f\n", radius);

    return 0;
}
Result:
Radius using double: 5.000000
```

ચાલો, આપેલા ક્ષેત્રફળ (area) અને પાઇ π (pi) ના મૂલ્યનો ઉપયોગ કરીને વર્તુળની ત્રિજ્યા શોધવા માટેના ઉપરના C પ્રોગ્રામને સમજીએ. પ્રોગ્રામ *sqrt()* ફંક્શનને ઉપયોગ કરવા માટે <*math.h*> ને સામેલ કરીને શરૂ થાય છે. યાદ રાખો કે <*math.h*> એ ગાણિતિક હેડર ફાઇલ છે, જે વર્ગમૂળ (square roots) ની ગણતરી માટે જરૂરી ફંક્શન ધરાવે છે. *double* પ્રકારના ત્રણ ચલ ઘોષિત કરવામાં આવ્યા છેઃ area, pi અને radius. area નું મૂલ્ય 78.5 તરીકે આપવામાં આવ્યું છે, અને pi ને 3.14 મૂલ્ય આપેલ છે. ત્રિજ્યા શોધવા માટે, radius = sqrt(area/pi); સૂત્રનો ઉપયોગ કરવામાં આવ્યો છે, જે ક્ષેત્રફળને pi વડે ભાગે છે અને પછી

Get More Learning Materials Here : CLICK HERE www.studentbro.in

પરિણામનું વર્ગમૂળ લે છે. *sqrt()* ફંક્શન આ વર્ગમૂળની પ્રક્રિયા કરે છે. છેલ્લે, પરિણામને સ્ક્રીન પર %f ફોર્મેટ સ્પેસિફાયરનો ઉપયોગ કરીને પ્રિન્ટ કરવામાં આવેલ છે, જે ખાસ કરીને *double* ટાઈપના ચલનું મૂલ્ય દર્શાવવા માટે વપરાય છે.

char ડેટા પ્રકારો (char Data Types)

char ડેટા પ્રકારનો ઉપયોગ અક્ષરો, અંકો અથવા પ્રતીકો જેવા એક જ અક્ષર (single characters) સંગ્રહવા માટે થાય છે, જેને હંમેશા સિંગલ અવતરણ ચિહ્નો (‘’) માં લખવામાં આવે છે. ઉદાહરણ તરીકે, જ્યારે આપણે char grade = ‘A’; લખીએ છીએ, ત્યારે આપણે grade નામનો એક ચલ બનાવી રહ્યા છીએ જે ‘A’ અક્ષર ધરાવે છે. આ સરળ છતાં મહત્ત્વપૂર્ણ ડેટા ટાઈપ આપણને વ્યક્તિગત કીબોર્ડ અક્ષરો સાથે કામ કરવાની મંજૂરી આપે છે, પછી ભલે આપણે કોઈના પ્રથમ અક્ષર (initial) અથવા વિરામચિહ્ન (punctuation mark) સંગ્રહી રહ્યા હોઈએ. *char* પ્રકાર ખાસ કરીને ટેક્સ્ટ ઈનપુટ, પાસવર્ડ અથવા અક્ષરો અને પ્રતીકોને મેનેજ કરવાની જરૂર હોય તેવા કોઈપણ પ્રોગ્રામ સાથે કામ કરવા માટે ઉપયોગી છે. પ્રોગ્રામિંગમાં, રેન્જ (Range) એટલે કે ચલનું લઘુત્તમ (minimum) થી મહત્તમ (maximum) મૂલ્ય, *signed char* ની રેન્જ -128 થી 127 સુધીની હોય છે, જ્યારે *unsigned char* ની રેન્જ 0 થી 255 સુધીની હોય છે. ટેબલ 7.4 *char* ડેટા પ્રકાર દર્શાવેલા છે. યાદ રાખો કે *char* ડેટા પ્રકારનો ઉપયોગ 8-બિટ *signed* અથવા *unsigned* પૂર્ણાંક (integers) તરીકે પણ થઈ શકે છે.

પ્રકાર (Type)	વર્ણન	ઉદાહરણ
char	એક જ અક્ષર સંગ્રહ કરે છે (કમ્પાઈલર પર આધાર રાખીને મૂળભૂત રીતે signed અથવા unsigned)	char letter=‘B’;
signed char	અક્ષરોને signed મૂલ્યો તરીકે સંગ્રહ કરે છે (-128 થી 127)	signed char sign = 'z';
unsigned char	અક્ષરોને unsigned મૂલ્યો તરીકે સંગ્રહ કરે છે (0 થી 255)	unsigned char symbol = '$';

ટેબલ 7.4 : char ડેટા પ્રકારો

ચાલો, આપણે એક ઉદાહરણ લઈએ.

```
/* C program to illustrate use of character.*/
#include <stdio.h>
int main()
{
    char grade = 'A';
    printf("Your grade is: %c", grade);

    return 0;
}

Result:
Your grade is: A
```

ચાલો ઉપરના પ્રોગ્રામને સમજીએ. અહીં grade નામનો એક *char* પ્રકારનો ચલ ઘોષિત કરવામાં આવ્યો છે અને તેમાં ‘A’ સંગ્રહવામાં આવ્યો છે. કેરેક્ટર હંમેશા single quotes (‘ ’)માં મુકવામાં આવે છે. પછી *printf()* ફંક્શનનો ઉપયોગ કરીને grade ચલમાં રહેલ અક્ષર પ્રિન્ટ કરવામાં આવે છે, જેમાં %c ફોર્મેટ સ્પેસિફાયર ઉપયોગ કરેલ છે, જે ખાસ કરીને અક્ષરો પ્રિન્ટ કરવા માટે વપરાય છે. અંતે, પ્રોગ્રામ return 0; વડે પૂર્ણ થાય છે, જે દર્શાવે છે કે પ્રોગ્રામ સફળતાપૂર્વક ચલાવવામાં આવ્યો છે.

Get More Learning Materials Here : CLICK HERE www.studentbro.in

ડિરાઈવ્ડ ડેટા પ્રકારો (Derived Data Types)

ડિરાઈવ્ડ (Derived) ડેટા પ્રકાર તે છે જે મૂળભૂત અથવા પ્રાથમિક ડેટા પ્રકાર પરથી બનાવેલ ડેટા પ્રકાર છે. તે અનેક મૂલ્યો અથવા જટિલ ડેટા સ્ટ્રક્ચર કાર્યક્ષમ રીતે સંગ્રહવા માટે વપરાય છે. ડિરાઈવ્ડ ડેટા ટાઈપ લવચીકતા (flexibility) પ્રદાન કરે છે અને વાસ્તવિક પ્રોગ્રામિંગમાં વ્યાપકપણે વપરાય છે, ખાસ કરીને અરે, મેમરી એડ્રેસ અને ગ્રુપ ડેટાને સંભાળવા માટે. ટેબલ 7.5માં ડિરાઈવ્ડ ડેટા ટાઈપની યાદી આપવામાં આવી છે.

પ્રકાર (Type)	વર્ણન	ઉદાહરણ
Array	એકસરખા ડેટા ટાઈપના મૂલ્યોનો સંગ્રહ, જે સતત (contiguous) મેમરીમાં સંગ્રહ થયેલ હોય છે.	int numbers[5] = {1, 2, 3, 4, 5};
Pointer	બીજા ચલના મેમરી એડ્રેસને સંગ્રહ કરે છે.	int *ptr; ptr = #

ટેબલ 7.5 : ડિરાઈવ્ડ ડેટા પ્રકારો

યુઝર-નિર્મિત ડેટા પ્રકારો (User-Defined Data Types)

યુઝર-નિર્મિત (User-defined) ડેટા પ્રકાર પ્રોગ્રામરને મૂળભૂત અથવા ડિરાઈવ્ડ (derived) પ્રકારના આધાર પર પોતાના ડેટા પ્રકારો બનાવવા દે છે. આ કસ્ટમ (ખાસ બનાવેલ) પ્રકાર જટિલ ડેટાને સુવ્યવસ્થિત કરવામાં અને કોડની વાંચનક્ષમતા તથા જાળવણી ક્ષમતા સુધારવામાં મદદરૂપ બને છે. C ભાષામાં મુખ્ય યુઝર-નિર્મિત ડેટા ટાઈપ છે: *struct* અને *union*. તેમની યાદી ટેબલ 7.6માં આપવામાં આવી છે.

પ્રકાર (Type)	વર્ણન	ઉદાહરણ

struct	વિવિધ પ્રકારના ચલોને એક જ નામ હેઠળ જૂથબદ્ધ કરે છે.	struct Student {
		int id;
		char name [20];
		};

union	એક જ મેમરી સ્થાનમાં (એક સમયે એક પ્રકારના) વિવિધ ડેટા ટાઈપ સંગ્રહવાની મંજૂરી આપે છે.	union Data {
		int i;
		float f;
		};

ટેબલ 7.6 : યુઝર-નિર્મિત ડેટા પ્રકારો

ચાલો એક ઉદાહરણ સાથે એરેને સમજીએ. એરે એકસરખા ડેટા પ્રકારની કિંમતોનો સંગ્રહ છે, જે સતત મેમરી સ્થાનોમાં સંગ્રહ થયેલ છે અને તેમને ઈન્ડેક્સ (index-સૂચક) નો ઉપયોગ કરીને મેળવી શકાય છે.

વાક્યરચના (Syntax):

```
data_type array_name[size];

Example: int marks[5] = {90, 80, 70, 60, 50};
```


Get More Learning Materials Here : CLICK HERE www.studentbro.in

```
/*C program to illustrate use of an array*/
#include <stdio.h>
int main()
{
    int numbers[5] = {10, 20, 30, 40, 50};
    printf("The elements of the array are:\n");
    printf("%d ", numbers[0]);
    printf("%d ", numbers[1]);
    printf("%d ", numbers[2]);
    printf("%d ", numbers[3]);
    printf("%d\n", numbers[4]);
    return 0;
}
Result:
The elements of the array are:
10 20 30 40 50
```

પ્રોગ્રામની શરૂઆતમાં numbers નામની એક પૂર્ણાંક સંખ્યાઓની એરે ઘોષિત કરવામાં આવી છે જેમાં 5 કિંમતો છે: 10, 20, 30, 40 અને 50. C ભાષામાં એરે આપણને એકસરખા પ્રકારના અનેક મૂલ્યોને એક જ ચલમાં સંગ્રહવાની સુવિધા આપે છે, જેમાં કિંમતોને ઈન્ડેક્સ (index) દ્વારા મેળવવામાં આવે છે, અને ઈન્ડેક્સની ગણતરી 0 થી શરૂ થાય છે. આ પ્રોગ્રામમાં એરેની દરેક કિંમતને તેના ઈન્ડેક્સ દ્વારા સીધી જ ઉપયોગમાં લીધેલ છે. ઉદાહરણ તરીકે, numbers[0] પ્રથમ કિંમત મેળવે છે, numbers[1] બીજી કિંમતને, અને આમ આગળ વધે છે. બધી કિંમતો પ્રિન્ટ કરવા માટે લૂપનો ઉપયોગ કરવાની જગ્યાએ, અહીં અલગ-અલગ *printf()* નો ઉપયોગ કરીને દરેક કિંમત પ્રિન્ટ કરવામાં આવી છે. આ રીત સરળ છે અને નાની એરે માટે યોગ્ય છે, પરંતુ મોટી એરે માટે અકાર્યક્ષમ બને છે. આ ઉદાહરણ શરૂઆતના શીખનારાઓને C પ્રોગ્રામિંગમાં એરેની ઘોષણા (declaration), પ્રારંભિકકરણ (initialization) અને અલગ-અલગ દરેક કિંમતોને કેવી રીતે જુદી-જુદી ઉપયોગમાં લેવી તે સમજવામાં મદદ કરે છે.

સ્ટોરેજ ક્લાસીસ અને તેની અગત્યતા (Storage Classes and Their Significance)

સ્ટોરેજ ક્લાસિસ ચલનો વ્યાપ (scope), જીવનકાળ (lifetime) અને લિંકેજ (linkage) નક્કી કરે છે. તે કમ્પાઇલરને પ્રોગ્રામની અંદર ચલની મેમરી અને ઉપલબ્ધતા કેવી રીતે સંભાળવી તે સમજવામાં મદદ કરે છે. C ભાષામાં સ્ટોરેજ ક્લાસિસના ચાર મુખ્ય પ્રકાર છે: auto, register, static, અને global. દરેક સ્ટોરેજ ક્લાસના પોતાના નિયમો હોય છે, જે ચલને ક્યાં ઉપલબ્ધ કરી શકાય અને તે મેમરીમાં કેટલો સમય ટકી રહે છે તે નક્કી કરે છે. ટેબલ 7.7 માં આ સ્ટોરેજ ક્લાસની યાદી આપવામાં આવી છે.

સ્ટોરેજ ક્લાસ	કીવર્ડ	વ્યાપ (scope) અને જીવનકાળ (life)	ઉદાહરણ
Automatic (local)	auto	બ્લોક/ફંક્શન માટે સ્થાનિક (local); બ્લોક/ફંક્શન સમાપ્ત થાય ત્યાં સુધી અસ્તિત્વ ધરાવે છે.	int a = 5;
Register	register	ફંક્શન માટે સ્થાનિક પરંતુ ઝડપ મેળવવા માટે તેને CPU રજિસ્ટરમાં સંગ્રહ કરવામાં આવે છે.	register int speed = 10;
Static	static	ફંક્શન માટે સ્થાનિક પરંતુ કોલ વચ્ચે તેનું અગાઉનું મૂલ્ય જાળવી રાખે છે.	static int count = 0;
Global	-	ગ્લોબલ (આખા પ્રોગ્રામમાં) વ્યાપ; બધા ફંક્શનની બહાર વ્યાખ્યાયિત	int total;

ટેબલ 7.7 : સ્ટોરેજ ક્લાસીસ અને તેની અગત્યતા

Get More Learning Materials Here : CLICK HERE www.studentbro.in

void ડેટા પ્રકાર (void Data Type)

C ભાષામાં *void* ડેટા પ્રકાર મૂલ્યના અભાવ (absence of value) ને દર્શાવે છે. તેને ખાલી ડેટા ટાઈપ (empty data type) તરીકે પણ ઓળખવામાં આવે છે. અન્ય ડેટા ટાઈપ જેમ કે *int, float* અથવા *char* ડેટા સંગ્રહે છે, જ્યારે *void* સૂચવે છે કે કોઈ ડેટા પરત અપાતો નથી, સ્વીકારાતો નથી, અથવા સંગ્રહ થતો નથી. તે મુખ્યત્વે એવા ફંક્શનમાં વપરાય છે જે કોઈ મૂલ્ય પરત (return) આપતા નથી અથવા પેરામીટર્સ સ્વીકારતા નથી, તેમજ *void* પોઈન્ટરમાં પણ તેનો ઉપયોગ થાય છે. આકૃતિ 7.8માં *void* ડેટા પ્રકારો આપેલા છે.

વાક્યરચના (Syntax):

```
void main()
{
// code
}
```

ઉપયોગ પરિસ્થિતિ (Use Case)	વર્ણન	ઉદાહરણ
void Function (no return)	ફંક્શન કોઈ કાર્ય કરે છે પરંતુ કંઈ પણ મૂલ્ય પરત આપતું નથી.	void greet(){ ... }
void Parameter (no input)	ફંક્શન કોઈ ઈનપુટ પેરામીટર (argument) લેતું નથી.	void show(void){...}
void Pointer	એવો પોઈન્ટર જે કોઈપણ પ્રકારના એડ્રેસને સંગ્રહ કરી શકે છે.	void *ptr;

ટેબલ 7.8 : void ડેટા પ્રકારો

સારાંશ

આ પ્રકરણમાં, આપણે C પ્રોગ્રામિંગમાં વપરાતા વિવિધ પ્રકારના ડેટા અને તે પ્રોગ્રામમાં માહિતીનું સંચાલન કરવામાં કેવી રીતે મદદ કરે છે, તે વિશે વિગતવાર સમજ્યા. આપણે મૂળભૂત ડેટા પ્રકારો (Primary Data Types) જેમ કે int, float, double અને char થી શરૂઆત કરી, જે અનુક્રમે પૂર્ણ સંખ્યાઓ, અપૂર્ણાંક (દશાંશ) સંખ્યાઓ અને અક્ષરોનો સંગ્રહ કરે છે. આપણે મૂલ્યોની શ્રેણી (range) ને નિયંત્રિત કરવા માટે આ પ્રકારોના signed અને unsigned પ્રકારનો ઉપયોગ કેવી રીતે કરવો તે શીખ્યા. આપણે ડિરાઈવ્ડ ડેટા પ્રકારો (Derived Data Types) વિશે પણ જાણ્યું, જેમાં એરે અને પોઈન્ટરનો સમાવેશ થાય છે, જે બહુવિધ મૂલ્યો અથવા મેમરી એડ્રેસને સંભાળવા માટે હાલના પ્રકારોમાંથી બનાવવામાં આવે છે. ત્યારબાદ, આપણે યુઝર-નિર્મિત ડેટા પ્રકારો (User-Defined Data Types) જેમ કે struct અને union નો અભ્યાસ કર્યો, જે પ્રોગ્રામરોને ડેટાને વ્યવસ્થિત કરવા માટે યુઝર-નિર્મિત ફોર્મેટ બનાવવાની મંજૂરી આપે છે. આ પ્રકરણમાં સ્ટોરેજ ક્લાસિસ (Storage Classes) જેમ કે auto, static, register, અને global પણ આવરી લેવામાં આવ્યા છે, જે ચલનો વ્યાપ (scope) અને જીવનકાળ (lifetime) નક્કી કરે છે. છેલ્લે, આપણે void ડેટા પ્રકાર વિશે શીખ્યા, જેનો ઉપયોગ ફંક્શન અને પોઈન્ટરમાં ''કોઈ મૂલ્ય નહીં'' (no value) દર્શાવવા માટે થાય છે.

સ્વાધ્યાય

1. C પ્રોગ્રામિંગમાં ડેટા પ્રકારોનો હેતુ શું છે?
2. C માં ડેટા પ્રકારોની ચાર મુખ્ય શ્રેણીઓનાં નામ આપો.
3. પૂર્ણાંક (int) ડેટા પ્રકારની વ્યાખ્યા આપો અને એક ઉદાહરણ આપો.

Get More Learning Materials Here : CLICK HERE www.studentbro.in

4. float અને double ડેટા પ્રકારોને ઉદાહરણ સાથે સમજાવો.
5. C પ્રોગ્રામિંગમાં char (અક્ષર) ના ઉપયોગની સમજૂતી આપો.
6. યુઝર-નિર્મિત ડેટા પ્રકાર (User-Defined Data Type) શું છે? કોઈપણ બે પ્રકારના નામ આપો.
7. C માં ડિરાઈવ્ડ ડેટા પ્રકારો (Derived Data Types) શું છે? એક ઉદાહરણ આપો.
8. C માં void ડેટા પ્રકાર શું છે?
9. C માં ચાર સ્ટોરેજ ક્લાસ (Storage Classes) ની યાદી બનાવો અને તેમાંથી એકનું વર્ણન કરો.
10. signed int અને unsigned int વચ્ચે શું તફાવત છે?

11. સાચું કે ખોટું જણાવો.

(1) char ડેટા પ્રકાર એક સમયે એક કરતાં વધારે અક્ષરો સંગ્રહ કરી શકે છે.
(2) C પ્રોગ્રામમાં main() ફંક્શન વૈકલ્પિક છે.
(3) unsigned int ધન અને ઋણ બંને સંખ્યાઓ સંગ્રહ કરી શકે છે.
(4) C માં એરે (Arrays) એ ડિરાઈવ્ડ ડેટા પ્રકારો (Derived Data Types) નો ભાગ છે.
(5) સ્ટોરેજ ક્લાસ (Storage Class) મેમરીમાં ચલના જીવનકાળ (life) ને અસર કરે છે.

12. ખાલી જગ્યાઓ પૂરો.

(1) _____ ડેટા પ્રકારનો ઉપયોગ પૂર્ણ સંખ્યાઓ સંગ્રહ કરવા માટે થાય છે.
(2) _____ સ્ટોરેજ ક્લાસ ફંક્શન કોલ્સ વચ્ચે તેનું મૂલ્ય જાળવી રાખે છે.
(3) જે ફંક્શન કોઈ મૂલ્ય પાછું નથી આપતું તે _____ ડેટા પ્રકાર સાથે ઘોષિત કરવામાં આવે છે.
(4) એરે _____ ડેટા પ્રકારના મૂલ્યોનો સંગ્રહ કરે છે.
(5) _____ ડેટા પ્રકારનો ઉપયોગ એક જ અક્ષર સંગ્રહ કરવા માટે થાય છે.

13. બહુવિકલ્પી પ્રશ્નો. સૌથી યોગ્ય જવાબ પસંદ કરો.

(1) નીચેનામાંથી ક્યો ડેટા પ્રકાર દશાંશ સંખ્યાઓ સંગ્રહ કરવા ઉપયોગી છે?
(a) int (b) char (c) float (d) void

(2) નીચેનામાંથી ક્યો યુઝર-નિર્મિત ડેટા પ્રકાર છે?
(a) int (b) float (c) union (d) char

(3) નીચેનામાંથી ક્યો કીવર્ડ ચલને સ્થાનિક વ્યાપ (local scope) સાથે વ્યાખ્યાયિત કરે છે?
(a) auto (b) register (c) static (d) global

(4) નીચેનામાંથી ક્યું ફોર્મેટ સ્પેસિફાયર signed int માટે વપરાય છે?
(a) %u (b) %f (c) %d (d) %c

(5) નીચેનામાંથી ક્યો ડેટા પ્રકાર "કોઈ મૂલ્ય નહિ" દર્શાવે છે?
(a) void (b) int (c) char (d) float

(6) નીચેનામાંથી ક્યો ડેટા પ્રકાર એક અક્ષર સંગ્રહ કરવા ઉપયોગી છે?
(a) float (b) char (c) int (d) double

Get More Learning Materials Here : CLICK HERE www.studentbro.in

(7) નીચેનામાંથી ક્યો ડિરાઇવ્ડ ડેટા પ્રકાર મેમરી એડ્રેસ સંગ્રહ કરવા વપરાય છે?

(a) Array (b) Pointer (c) Struct (d) Union

(8) C માં float ડેટા પ્રકારની સાઈઝ કેટલી હોય છે?

(a) 2 bytes (b) 4 bytes (c) 8 bytes (d) 10 bytes

(9) CPU માં ઝડપી ઉપયોગ માટે ક્યો સ્ટોરેજ ક્લાસ ઉપયોગી છે?

(a) static (b) register (c) global (d) auto

(10) નીચેનામાંથી ક્યો યુઝર-નિર્મિત ડેટા પ્રકાર એકથી વધારે અલગ-અલગ પ્રકારના ચલો ધરાવી શકે?

(a) enum (b) struct (c) int (d) float

પ્રાયોગિક સ્વાધ્યાય

1. int અને unsigned int ના ઉપયોગનું નિદર્શન કરતો C પ્રોગ્રામ લખો.
2. બે float સંખ્યાઓનો ગુણાકાર કરતો C પ્રોગ્રામ લખો.
3. *printf()* ફંક્શનનો ઉપયોગ કરીને વિદ્યાર્થીની વિગતો દર્શાવતો C પ્રોગ્રામ લખો.
4. અરેનો ઉપયોગ કરીને 5 સંખ્યાઓનો સંગ્રહ અને પ્રિન્ટ કરતો પ્રોગ્રામ લખો.
5. તમારી શાળાનું સંપૂર્ણ સરનામું પ્રિન્ટ કરતો C પ્રોગ્રામ લખો.

Get More Learning Materials Here : CLICK HERE www.studentbro.in