

Programming with C and C++

CSC-101 (Lecture 07)

Dr. R. Balasubramanian
Professor

Department of Computer Science and Engineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Roorkee
Roorkee 247 667

bala@cs.iitr.ac.in

<https://faculty.iitr.ac.in/cs/bala/>



Linux Environment



🔄 cocalc.com/projects/4a4e8c0c-0df0-421a-9f93-f40afd01f705/files/prog1.c#line=1

Projects ■ CProg 📶 × + 🛒 Help

📄 Multi.c × 📄 prog1.c × 📄 2023-06-18-file-1.term ×

■ Stopped Project is available and ready to try to run.
▶ Start project

🔗 Source Code 🔄 🔄 🔄 📄 ⌵ ... 🔗 Code ⌵ ☰ ☐ 🔍 × 🏠 📖 🔄 — + 📶 ... 🔍

📄 prog1.c

```
1 ▾ /* Online C Compiler and Editor */
2 ▾ #include <stdio.h>
3
4 int main()
5 ▾ {
6     printf("Hello, World!\n");
7
8     return 0;
9 }
10
```

```
~$ gcc prog1.c
~$ ./a.out
Hello, World!
~$ █
```

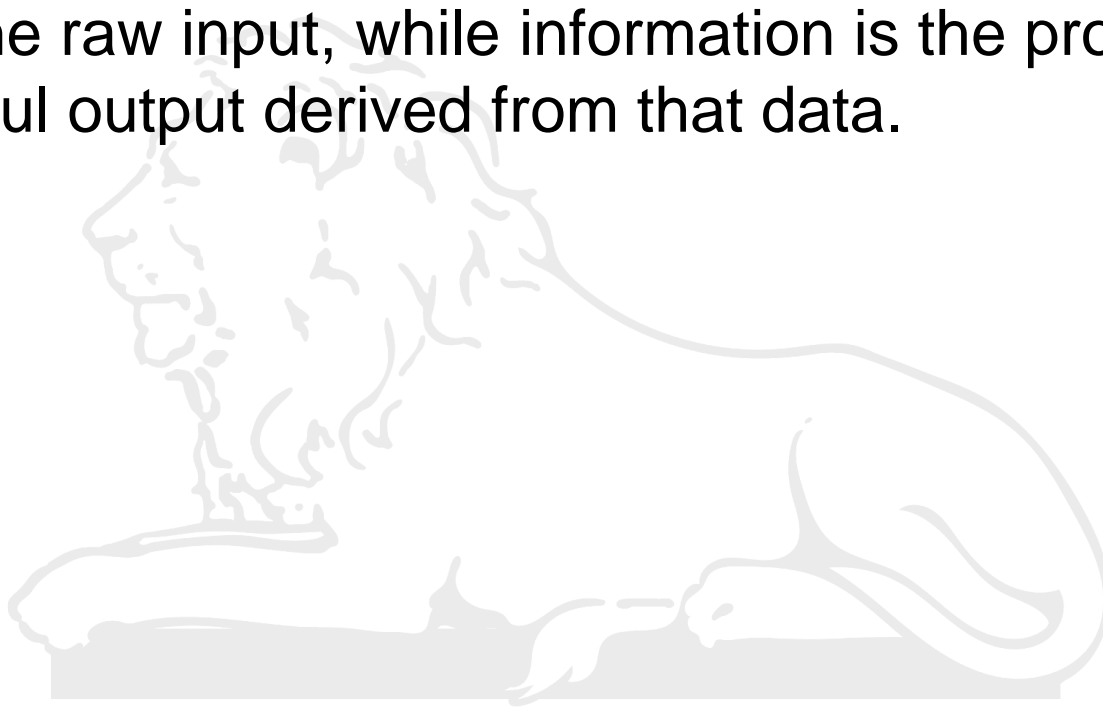
<https://cocalc.com/>

Linux commands



- ▶ ~\$ gcc hello1.c
- ▶ ~\$ vi sample.c
- ▶ Command i for inserting text
- ▶ Press Esc and then use colon : then type wq for write and quit
- ▶ ~\$ ls for list the files
- ▶ gcc stands for GNU Compiler Collections
- ▶ GNU stands for GNU's Not Unix
- ▶ <https://www.javatpoint.com/linux-commands>
- ▶ <https://www.geeksforgeeks.org/linux-commands/>

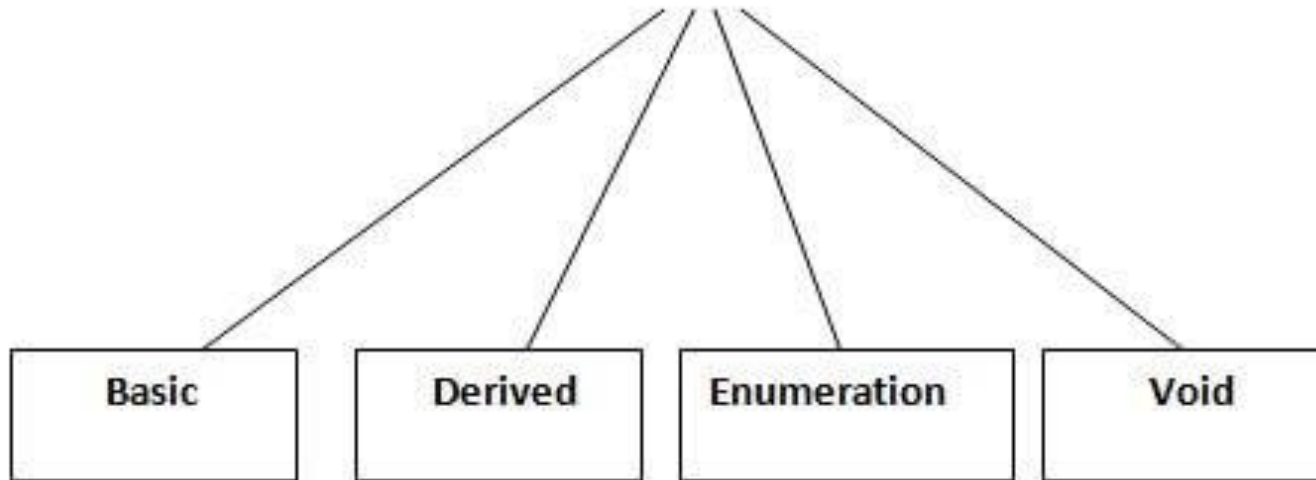
- ▶ Difference between Data and Information
- ▶ Data is the raw input, while information is the processed and meaningful output derived from that data.



Data Types in C



Data Types in C



Integer Types



Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or - 2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	8 bytes or (4 bytes for 32 bit OS)	-9223372036854775808 to 9223372036854775807
unsigned long	8 bytes	0 to 18446744073709551615



```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <float.h>

int main() {

    printf("CHAR_BIT      :    %d\n", CHAR_BIT);
    printf("CHAR_MAX      :    %d\n", CHAR_MAX);
    printf("CHAR_MIN      :    %d\n", CHAR_MIN);
    printf("INT_MAX       :    %d\n", INT_MAX);
    printf("INT_MIN       :    %d\n", INT_MIN);
    printf("LONG_MAX      :    %ld\n", (long) LONG_MAX);
    printf("LONG_MIN      :    %ld\n", (long) LONG_MIN);
    printf("SCHAR_MAX     :    %d\n", SCHAR_MAX);
    printf("SCHAR_MIN     :    %d\n", SCHAR_MIN);
```

```
printf("SHRT_MAX      :    %d\n", SHRT_MAX);  
printf("SHRT_MIN      :    %d\n", SHRT_MIN);  
printf("UCHAR_MAX     :    %d\n", UCHAR_MAX);  
printf("UINT_MAX       :    %u\n", (unsigned int) UINT_MAX);  
printf("ULONG_MAX      :    %lu\n", (unsigned long) ULONG_MAX);  
printf("USHRT_MAX      :    %d\n", (unsigned short) USHRT_MAX);  
  
return 0;  
}
```

<https://ideone.com/BUospa>

ANSI - American National Standards Institute

</> source code

```
1  #include <stdio.h>
2
3  int main(void) {
4      // your code goes here
5      int a=2147483647;
6      int b=2147483647;
7      int c= a+b;
8      printf("sum=%d", c);
9      return 0;
10 }
11
```

<https://ideone.com/jwrxmn>

Floating Point Types



Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places



```
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #include <limits.h>
4.  #include <float.h>
5.
6.  int main() {
7.
8.      printf("Storage size for float : %d \n", sizeof(float));
9.      printf("FLT_MAX      :  %g\n", (float) FLT_MAX);
10.     printf("FLT_MIN      :  %g\n", (float) FLT_MIN);
11.     printf("-FLT_MAX     :  %g\n", (float) -FLT_MAX);
12.     printf("-FLT_MIN     :  %g\n", (float) -FLT_MIN);
13.     printf("DBL_MAX      :  %g\n", (double) DBL_MAX);
14.     printf("DBL_MIN      :  %g\n", (double) DBL_MIN);
15.     printf("-DBL_MAX     :  %g\n", (double) -DBL_MAX);
16.     printf("Precision value: %d\n", FLT_DIG );
17.
18.     return 0;
19. }
```

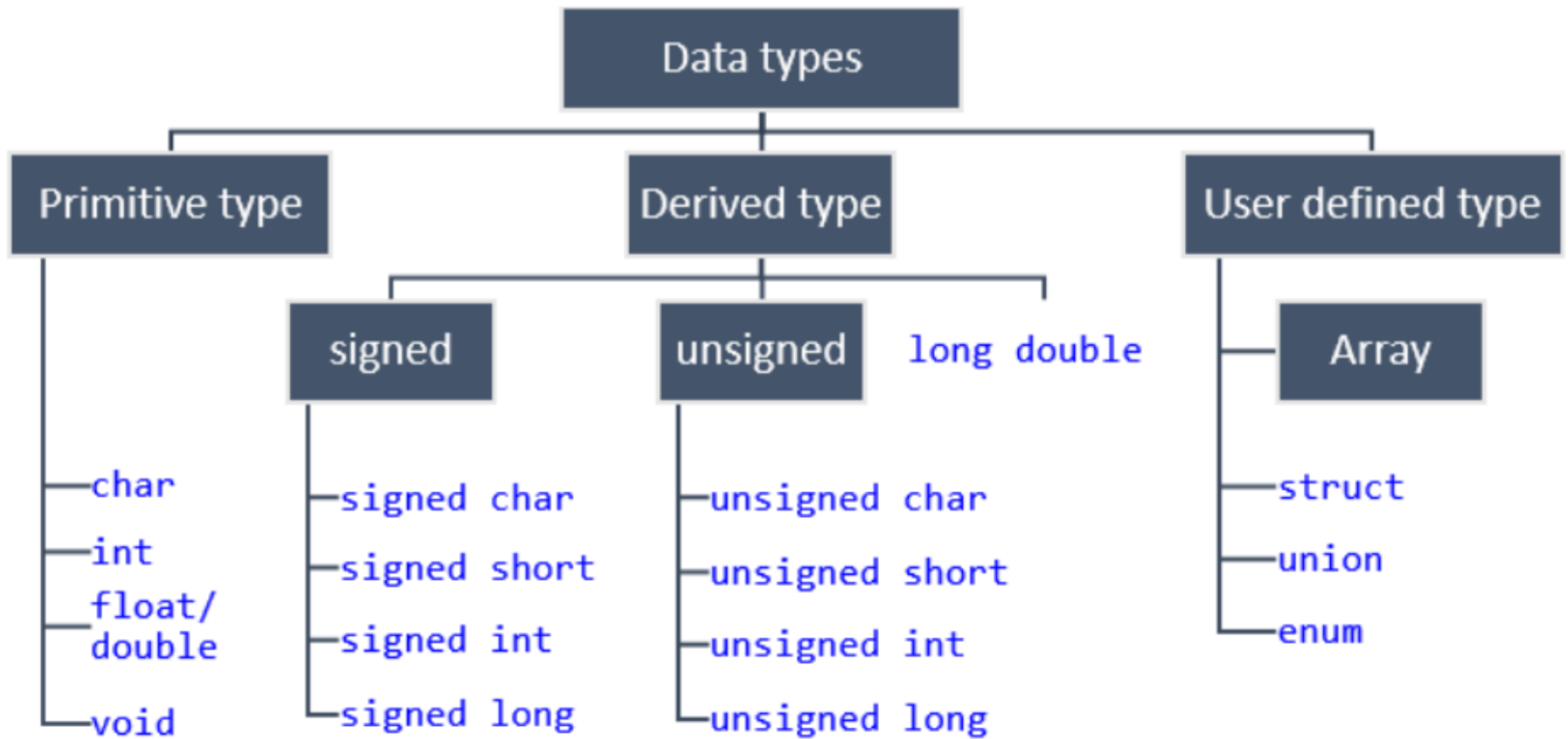


```
1.  #include <stdio.h>
2.
3.  int main()
4.  {
5.      char a = 'a';
6.      char c;
7.
8.      printf("Value of a: %c\n", a);
9.
10.     a++;
11.     printf("Value of a after increment is: %c\n", a);
12.
13.     // c is assigned ASCII values
14.     // which corresponds to the
15.     // character 'c'
16.     // a-->97 b-->98 c-->99
17.     // here c will be printed
18.     c = 99;
19.
20.     printf("Value of c: %c", c);
21.
22.     return 0;
23. }
```

<https://ideone.com/3B2Dh6>

- ▶ ASSCI stands for American Standard Code for Information Interchange





Type casing

