LEARN PROGRAMMING

STUDY GROUP - SESSION #3

Weekly: Wednesday 19:15 to 22:15
R307 G29

What is OUR PLAN?

- **Programming basic concepts**
- **Problem solving**
- **Pseudocode**
- **Flowcharts**
- Algorithms
- Syntax
- Compiling
- Debugging
- Bug fixing
- PRACTICE..PRACTICE..PRACTICE

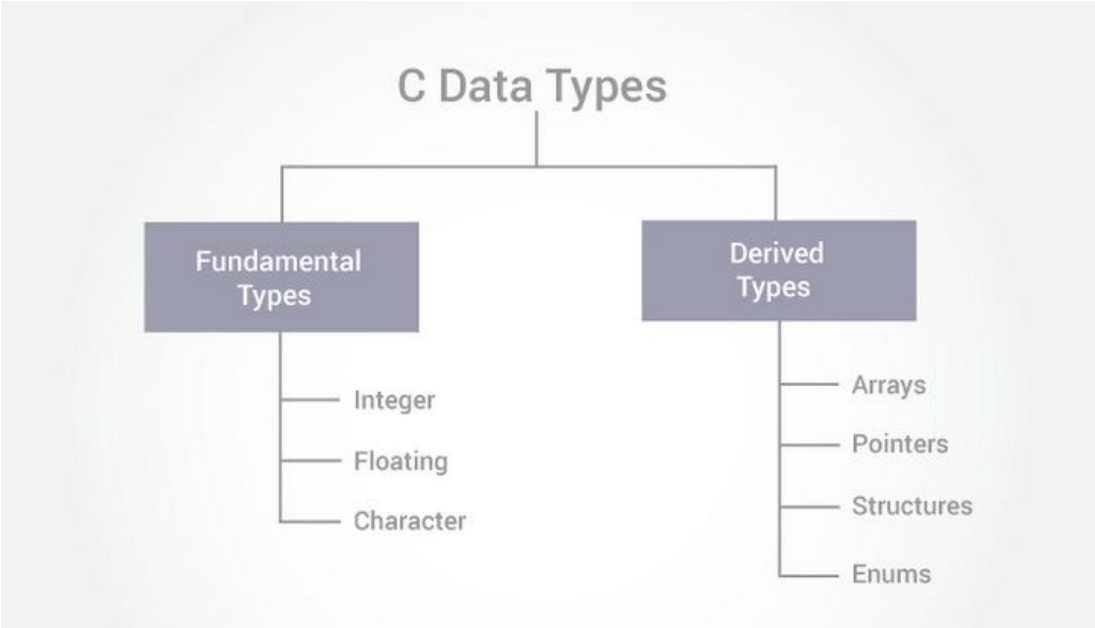
Data Types

Data types tells the processor what to **expect** as **value** and how much **memory** space should be reserved.

e.g A variable ‘X’ is declared as *int*, this simply tells the computer that reserve a space for ‘X’ variable so that it can simply hold the value say ‘30’.

1. Integer
2. Character
3. Floating

Data Types



https://www.programiz.com/c-programming/c-data-types

Data Representation

When information is put into a computer it is translated into a series of 1s and 0s it is called binary code.

This code is made up of only two digits “1” and “0.” These are called binary digits or bits. Eight bits are arranged into a pattern, which the computer interprets as a **byte** of information.

0 =	0
1 =	1
2 =	10
3 =	11
4 =	100
5 =	101
6 =	110
7 =	111
8 =	1000
9 =	1001
10 =	1010

11 =	1011
12 =	1100
13 =	1101
14 =	1110
15 =	1111
16 =	10000
17 =	10001
18 =	10010
19 =	10011
20 =	10100
…	

Data Representation

- 8 Bits = 1 Bytes
- A byte, can represent 256 (2^8) values ranging from 0 to 255, as shown here:
- 0 = 00000000
- 1 = 00000001
- 2 = 00000010
- ...
- 254 = 11111110
- 255 = 11111111

Bit #	7	6	5	4	3	2	1	0
Value	0	0	0	0	1	0	1	0

Example for value 10

```
0 =     0
1 =     1
2 =    10
3 =    11
4 =   100
5 =   101
6 =   110
7 =   111
8 =  1000
9 =  1001
10 =  1010
```

```
11 =  1011
12 =  1100
13 =  1101
14 =  1110
15 =  1111
16 = 10000
17 = 10001
18 = 10010
19 = 10011
20 = 10100
…
```

DATA REPRESENTATION

Memory Capacity Conversion Chart

Term (Abbreviation)	Approximate Size
Byte (B)	8 bits
Kilobyte (KB)	1024 bytes / 10^3 bytes
Megabyte (MB)	1024 KB / 10^6 bytes
Gigabyte (GB)	1024 MB / 10^9 bytes
Terabyte (TB)	1024 GB / 10^{12} bytes
Petabyte (PB)	1024 TB / 10^{15} bytes
Exabyte (EB)	1024 PB / 10^{18} bytes
Zettabyte (ZB)	1024 EB / 10^{21} bytes
Yottabyte (YB)	1024 ZB / 10^{24} bytes

```
0 =     0
1 =     1
2 =    10
3 =    11
4 =   100
5 =   101
6 =   110
7 =   111
8 =  1000
9 =  1001
10 =  1010
```

```
11 =  1011
12 =  1100
13 =  1101
14 =  1110
15 =  1111
16 = 10000
17 = 10001
18 = 10010
19 = 10011
20 = 10100
…
```

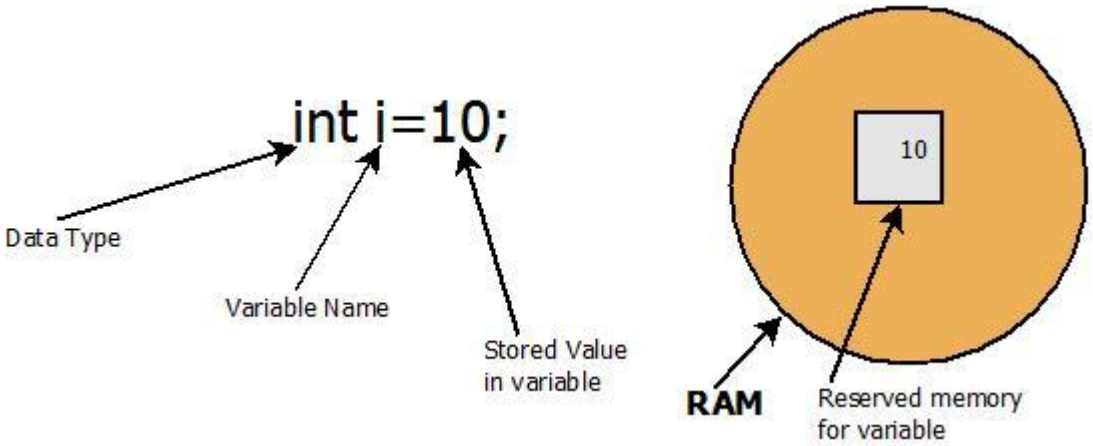
ASCII TABLE

Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

http://web.alfredstate.edu/weimandn/miscellaneous/ascii/ASCII%20Conversion%20Chart.pdf

VARIABLE


int j=10;
Data Type
Variable Name
Stored Value
in variable
10
RAM
Reserved memory
for variable

Data Types in C

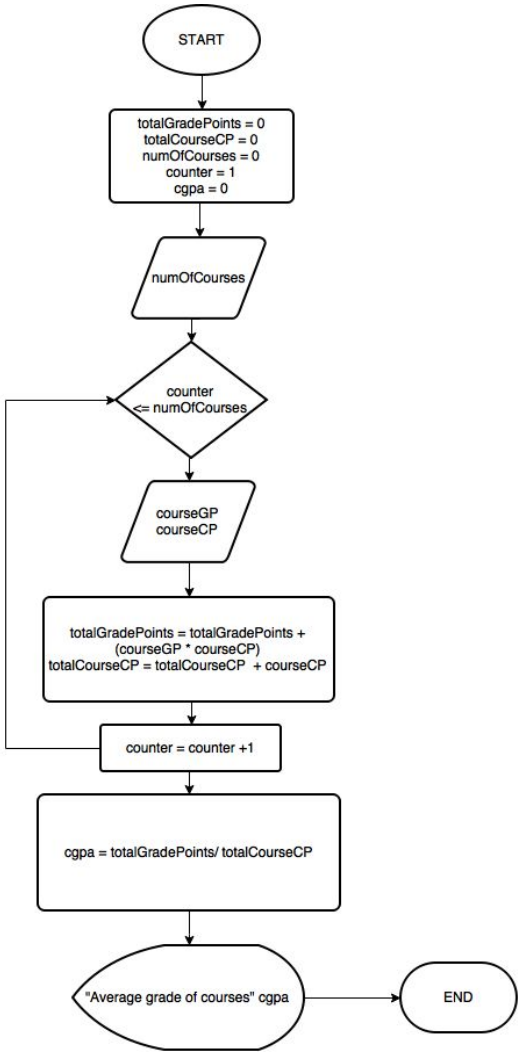
Type	Bytes	Values
int	2 or 4	-32, 768 to 32, 767
unsigned int	2 or 4	0 to 65, 535
signed int	2 or 4	-32, 767 to 32, 767
short int	2	-32, 767 to 32,767
unsinged short int	2	0 to 65, 535
signed short int	2	-32, 767 to 32, 767
long int	4	-2,147,483,647 to 2,147,483,647
signed long int	4	-2,147,483,647 to 2,147,483,647
unsigned long int	4	0 to 4, 294,967,294

PSEUDOCODE (CGPA CALCULATOR)

```
BEGIN
INITIALISE counter =1
INITIALISE totalGradePoints = 0
INITIALISE totalCreditPoints = 0
INITIALISE numOfCourses = 0
INITIALISE cgpa = 0
READ numOfCourses
```

```
WHILE counter<=numOfCourses
    READ courseGP
    READ courseCP
    totalGradePoints += (courseGP*courseCP)
    totalCreditPoints +=courseCP
    counter +=1
ENDWHILE
cgpa = totalGradePoints/totalCreditPoints
WRITE “Average grade is: ” cgpa
END
```

FLOW CHART (EXAMPLE)



SOLUTION

```cpp
#include <iostream> // header files for input/output
using namespace std;

int main() // function to start from
{ //start of function body


    //start here program code
   cout << "Hello World to Asema" << endl;

   //Writing the CGPA Calculator program here
   int counter = 1;
   float totalGP = 0;
   int totalCP=0;
   int numOfCourses=0;

   float courseGP=0;
   int courseCP=0;

   float cgpa=0;

   cout<<"Please enter your # of courses:";
   cin>>numOfCourses;

   while(counter<=numOfCourses)
   {

     cout<< "Course: " <<counter<<endl;

     cout<<"Please enter your course GP:";
     cin>>courseGP;

     //condition
     if(courseGP < 5)
     {
        cout<<"Please enter your course CP:";
        cin>>courseCP;

        totalGP = totalGP + (courseGP * courseCP);
        totalCP= totalCP + courseCP;
     }
     counter = counter + 1; // counter = counter + 1; // counter +=1;

   }
    cgpa = totalGP/totalCP;
    cout<< "CGPA: " <<cgpa <<endl;

   //End CGPA Calculator
   //End before this
   return 0;
}// body ends
```

TASK FOR TODAY?

1. Write a program that prints the numbers **from 1 to 100.** But for multiples of **three print "Fizz"** instead of the number and for the multiples of **five print "Buzz".** For numbers which are multiples of **both three and five print "FizzBuzz".**

TASK FOR TODAY? (SOLUTION)

ONLINE TOOL:

https://www.tutorialspoint.com/compile_cpp_online.php

```cpp
#include <iostream>
using namespace std;

int main()
{
        cout <<"Fizz Buzz Problem\n";

        for(int i=1; i<=100; i++)
        {
            if (i % 3 == 0 && i % 5 == 0)
            {
                cout << i <<" FizzBuzz\n";
            }
            else
            {
                if (i % 3 == 0)
                {
                    cout << i <<" Fizz\n";
                }
                if (i % 5 == 0)
                {
                    cout<< i <<" Buzz\n";
                }
            }
        }

    return 0;
}
```

C++ example

(Solution from Session)

USED ONLINE TOOL:

https://www.tutorialspoint.com/compile_cpp_online.php

```cpp
#include <iostream>

using namespace std;

int main()
{
   //solution
   int counter=1;

   while(counter<=100)
   {

      //Sortof solution
    /*   if(counter%3==0)
            cout<<counter<<" : Fizz";
      if(counter%5==0)
            cout<<counter<<" : Buzz"<<endl; */


      //Perfect solution
      if(counter%3==0 && counter%5==0)
         cout<<counter<<"FizzBuzz"<<endl;
       else{
        if(counter%3==0)
            cout<<counter<<" : Fizz"<<endl;
        if(counter%5==0)
            cout<<counter<<" : Buzz"<<endl;
       else
            cout<<counter<<endl;
       }


      //Simple solution, but doesnt fulfill all conditions
     /*
     if(counter%3==0)
       cout<<counter<<" : Fizz"<<endl;
     if(counter%5==0)
       cout<<counter<<" : Buzz"<<endl;
     else if(counter%3==0 && counter%5==0)
       cout<<"FizzBuzz"<<endl;
    else
       cout<<counter<<endl;
       */
       counter++;
   }

   return 0;
}
```

THANK YOU

Let's get started

TOOL & TECH

Integrated Development Environment (IDE)

JETBRAINS: https://www.jetbrains.com/student/

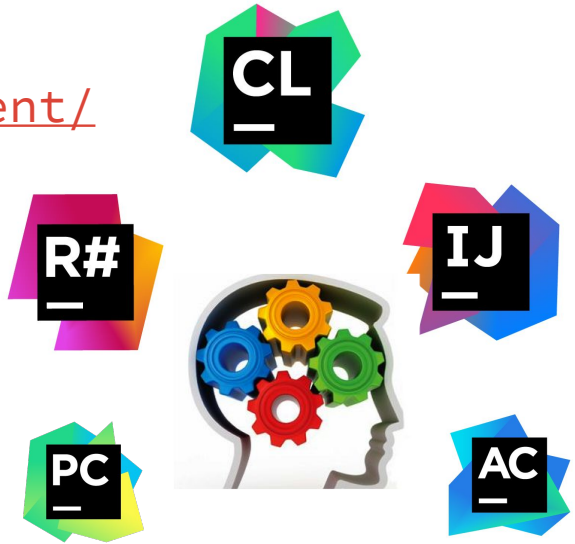
CLION - C/C++ (Install THIS)

Resharper - C#

IntelliJ IDEA - JAVA

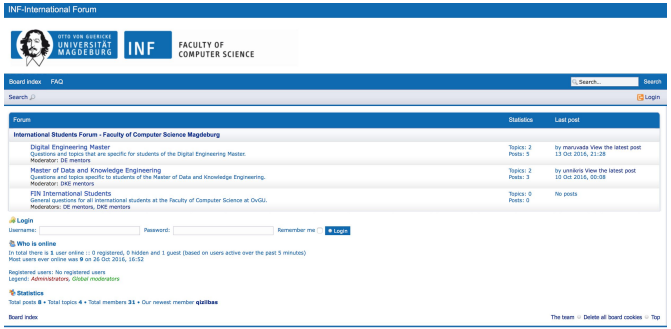
PyCharm - Python

AppCode MacOSX/iOS



HOW TO REACH US?

- FORUM for International students?
- Link: https://iif.cs.uni-magdeburg.de/index.php
- “Study group programming”
- Login with your university ID.
- Post your questions on FORUM.
- Moderators will reply.



OFFICIAL COMMUNICATION ---> FORUM ONLY