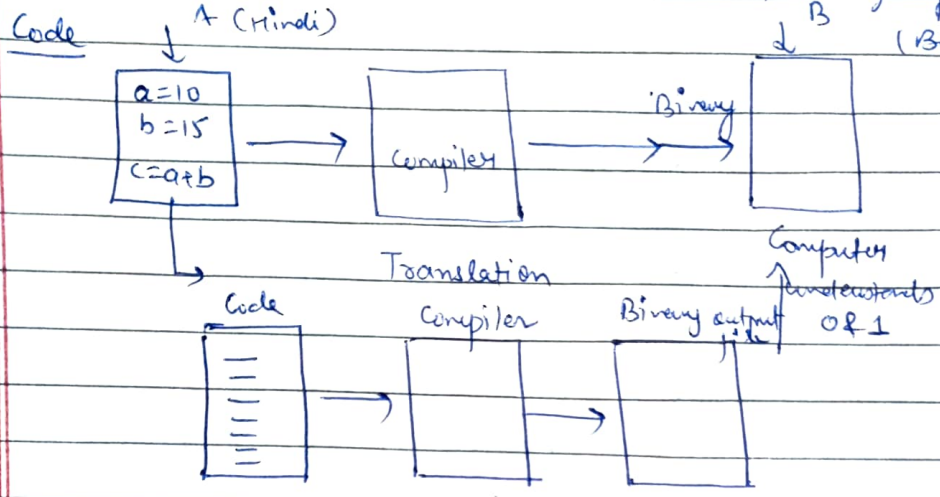
Dt 7th Dec 2022
Pg

Getting Started Module

For A and B communication I can't use different languages



Compile Time errors → Syntactical

a , b → a+b

Run Time Errors → Logical

IDE → Integrated Development Environment

Compile & Run Together → Doing by itself

Extension of cpp file → .cpp

Start Block in Flowcharts ↔ main()

int main () {

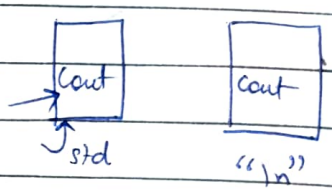
cout << "Hello World" << endl; cout << "\n";

}

insertion operator

#include < iostream >

using namespace std;



; → full stop (.)

To print something as it on console put in double quotes (" ")

"\n" → endl.

WOW!

Dt.
Pg.

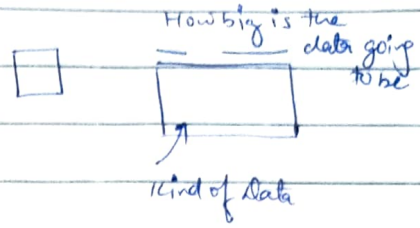
Variables and Data Types

(1)

```
int main(){
    int a=10;
    int b=20;
    int c=a+b;
    cout<<c<<endl;
}
```

a → 10 → 4 bytes

int b → number; how big is the data



int → number
→ storage required
→ 4 bytes

(2)

```
int main(){
    char d='x';
    cout<<d<<endl;
}
```

→ 1 byte

(3)

```
int main(){
    float f=1.23;
    double g=1.234;
    cout<<f<<" "<<g<<endl;
}
```

→ 4 bytes
→ 8 bytes

(4)

```
int main(){
    bool f=true;
    cout<<f<<endl;
}
```

→ 1 bytes → prints 0 & 1 as output, not true or false

(5) sizeof(f) → tells the amount of data that it can store.

(6) No two same name of variables in same scope;

```
float b=15;
int b=15;
```

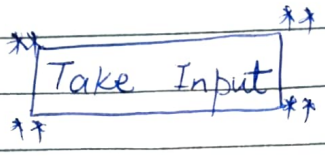
✗ Not allowed in same scope

WOW!

Dt
Pg

Variable Naming Rules

1) It can contain numbers (0-9), alphabets (A...Z, a...z) and underscore (_)
2) It cannot start with a number but can start with an underscore.
3) All variables contain garbage value before initialization.
4) The output of a boolean variable is either 0 or 1.




```
int main(){
    int a,b;
    cin>>a>>b;
    cout<< a+b;
    }
```

int a; 4bytes
a → not clean

int b;
b
same storage at some point of time of a garbage value

How is Data Stored?

Binary → output

int a=100;

a
100
bytes
⇒
Binary
4bytes

int a=259;

256 + 2 + 1

0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

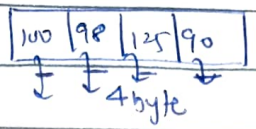
char d='y'; → (All characters possess ASCII value)

1 byte

~~ASCII Table~~

65-92 → A-Z
97-122 → a-z
48-57 → 0-9

WOW!



Might be char, one integer

Depends on what is stored & how to interpret it.

2.35

↓

10. ——

$1.(xxxx) * 2^{e}$ ← exponent

mantissa

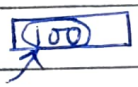
100.1101

$1.001101 * 2^{2}$

23.37

$2.337 * 2^{1}$

```
int main(){
    int a=100;
    char c=a;
    cout<<c<<endl;
}
```



→ Implicit Typecasting

```
c='4';
a=c;
cout<<a<<endl;
}
```

char + integer → integer

↓

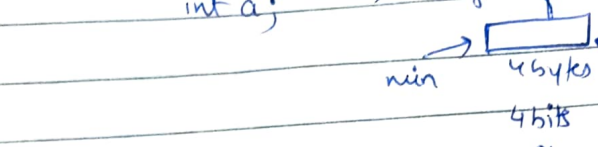
ASCII

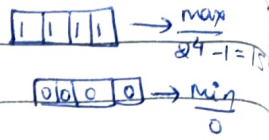
WOW!

Dt
Pg

How are Negative Numbers Stored?

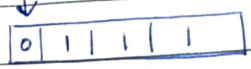
int a; → Range of a



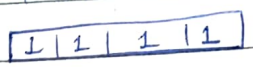


1's bit → Sign bit
→ 0 → +ive
→ 1 → -ive

-n



$max = 2^3 - 1 = 7$



min

$-7 - (2^{n-1} - 1)$

0 0 0 0 → +0

1 0 0 0 → -0

i) +ive number → Convert to binary

-ive no → ① Forget negative sign

② Convert num to binary

③ Take 2's complement

④ Store the result of Step-3

0 1 0 1

1's Complement → 1 0 1 0

+ 1

2's Complement → 1 0 1 1

-5 → 0101

1st → 1010

+1

1011 → 2's complement → (-ive number)

WOW!

1010

2's complement → 0101
+1
0101 →

32 bits
[11111 ... 101]

Range of Integers → $2^{n-1} - 1$
-2^{n-1}

$(-2^{n-1}) \ldots\ldots (2^{n-1} - 1)$

where n = no. of bits

Only positive numbers → float, double
floating point numbers
→ long, int

unsigned int a → All positives

Range → $-2^{n-1} \ldots 2^{n-1} - 1$

$0 \ldots\ldots 2^{n} - 1$

-128 127

Operators

```
int main(){
    int f;
    cout << "Enter Fah Value" << endl;
    cin>>f;
    int c = (f-32)*5/9;
    cout<<c<<endl;
}
```

5/9 * (f-32)
↓
0.55

int/int → int
float/int → float
double/int → double

WOW!

Dt
Pg

Arithmetic Opr

+, -, *, /, %

```
bool ans = (a==b);
cout << ans;
```

Relational Opr

> → Greater than

< → Lesser than

>= → Greater than or equal to

<= → Less than or equal to

== → equal to

!= → Not equal to

Logical Opr

&& → AND ALSO

|| → OR ALSO

! → NOT

F&&F → F	F||F → F
T&&F → F	F||T → T
F&&T → F	T||F → T
T&&T → T	T||T → T

Note Point → Modulus Operator cannot work with int or float (%)