Module 1!  
  
1972 Dennis Ritchie

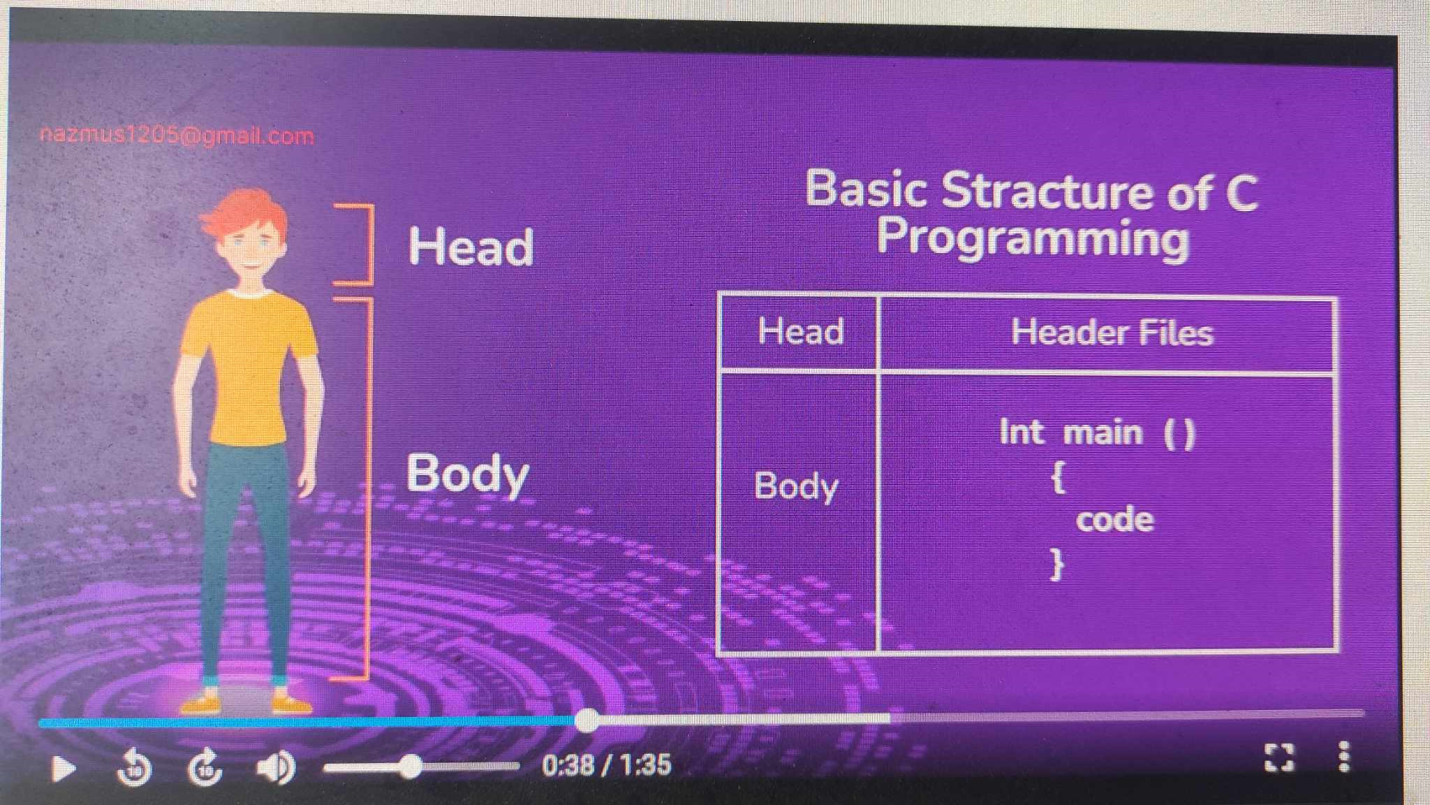
Mother of programming languages

First day topic:

1. Input
2. Print
3. Variable and data type
4. Data type limitations
5. Rules for variable names.

Question: where is github repo that he talked about??

1-1:



Summary:

Header file is the code that is hard to code and takes lot of time to code, and that’s why it is already given to make our coding life easy.

1-2;

stdio = standard input and output

1-3;

We can consider a robot,

We give him milk powder, tea powder,

Then he process

Finally he gives us Milk tea.

Here robot is working like a function we are giving some input, and after processing he is giving the expected output.

There are two kinds of functions:

1. User Defined Functions
2. Built in Functions

The function we start with is main function [ return\_type main(){} ]

Basic structure of function:

Return\_type name( )  
{  
 process;  
 return;  
}

Grammartical rules of code = syntax

Header file and Library file

<angle brackets>

1-4;

Special characters (ESCAPE)

1. \n – new line
2. \t – tab (4 spaces)

We can’t write \ single backslash  
We have to write \\ double backslash to print single backslash  
Same in case of %% to print %

1-5;

1. int - -100, 0, 100
2. float - -2.5, 5.46
3. char – ‘a’, ‘A’, ‘@’, ‘H’
4. bool – True or False

1-6;

int takes 4 bytes / 32 bits of space in memory

1 gb = 1024 mb  
1 mb = 1024 kb  
1 kb = 1024 byte  
1 byte = 8 bit

int rahim; // declare  
rahim = 100; // assign

int rahim = 100;// initialize = declare + assign

float takes 4 bytes / 32 bits too.

Format specifier:

1. %d = int
2. %f = float
3. %c = char

1-8;

& (ampersand) = Address of

1-9;

Data type limitations

int = 4 bytes, if we need more space then use long long int = 8 bytes  
float = 4 bytes, if we need more space then use double = 8 bytes

1-10;

Computer works with bit. 1 bit( 0 or 1 )

1 byte = 8 bit  
4 byte = 32 bit

Scale:  
0 0 0 0 0 0 1 = 1 bit (maximum limit)  
0 0 0 0 0 1 1 = 3 bit (maximum limit)  
64 32 16 8 4 2 1

1 bit representation:

0 and 1

2 bit representation:

0 0 = 0  
0 1 = 1  
1 0 = 2  
1 1 = 3

3 bit representation:

0 0 0 = 0  
0 0 1 = 1  
0 1 0 = 2  
0 1 1 = 3  
1 0 0 = 4  
1 0 1 = 5  
1 1 0 = 6  
1 1 1 = 7

4 bit representation:

0 0 0 0 = 0  
0 0 0 1 = 1  
0 0 1 0 = 2  
0 0 1 1 = 3  
0 1 0 0 = 4  
0 1 0 1 = 5  
0 1 1 0 = 6  
0 1 1 1 = 7  
1 0 0 0 = 8  
1 0 0 1 = 9  
1 0 1 0 = 10  
1 0 1 1 = 11  
1 1 0 0 = 12  
1 1 0 1 = 13  
1 1 1 0 = 14  
1 1 1 1 = 15

2 bit = 2^2 - 1 = 4-1 = 3  
3 bit = 2^3 - 1 = 8-1 = 7  
4 bit = 2^4 - 1 = 16-1 = 15

Since 4 byte = 32 bit

32 bit = 2^32 – 1 = 4,294,967,296-1 = 4,294,967,295 (10 digits)

So we can consider 1,000,000,000 (-10 ^9 to +10^9) as highest integer maximum possible value  
Here, we can take more till 4,294,967,295 but we won’t because compiler will start to behave unexpectedly. So 10^9 is safe and also easy to remember.

long long int takes 8 bytes which means 64 bits

64 bit = 2^64 – 1 = 18,446,744,073,709,551,616-1 = 18,446,744,073,709,551,615 (20 digits)

Here we can consider 10^18 as highest. (-10^18 to +10^18) for safety.

Important:  
float can contain 6-7 digits including [before and after .(doshomik)] less correct value in case of point  
(long float) double can contain 16 digits including [before point(.) and after(.)] more correct value in case of point.

1-11;

Rules for naming variables:

1. Variable name must start with a letter or underscore

rahim (alright)  
\_rahim (alright)  
@rahim (not alright)  
123rahim (not alright)

1. Variable name must contain letter, digits or underscore

rahim123 (alright)  
rahim\_123 (alright)  
rahim 123 (not alright)  
rahim, karim (not alright)

1. Keywords can’t be used as variables

int (not alright)  
for (not alright)  
float (not alright)