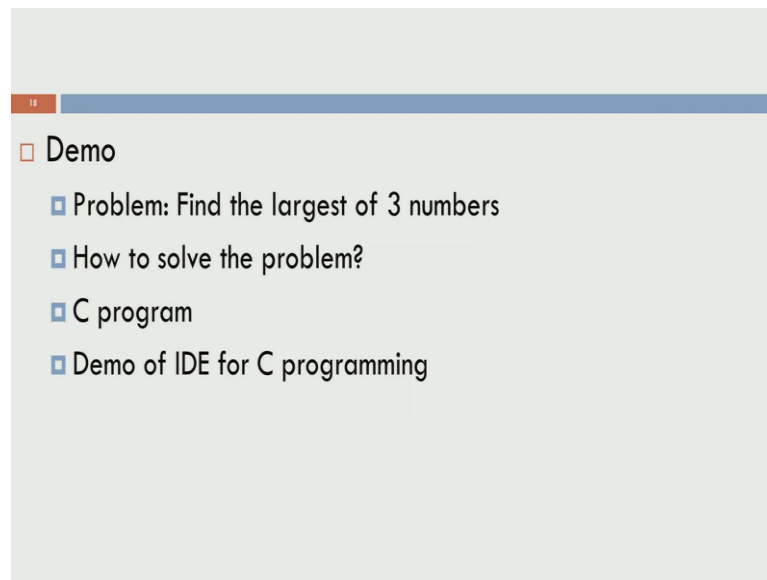


Programming, Data Structures and Algorithms
Prof. Shankar Balachandran
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 02
Lecture - 02
Problem Solving

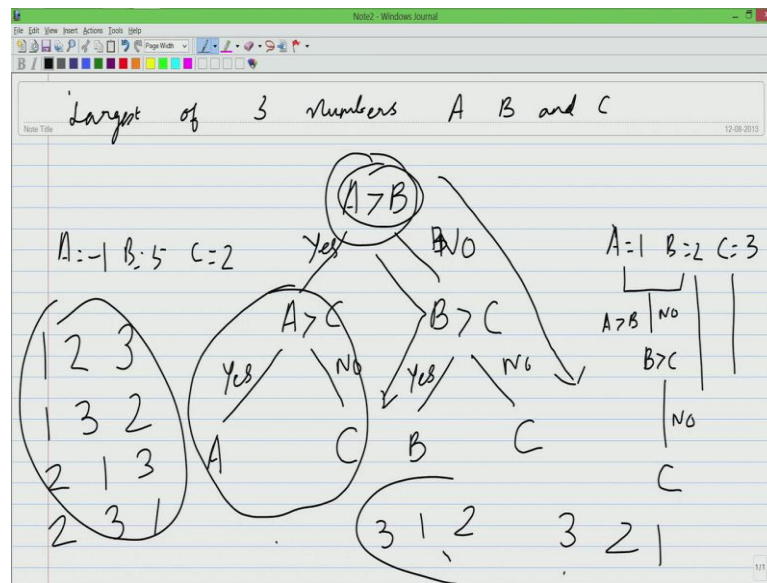
In module 2 we have looked at problem solving.

(Refer Slide Time: 00:16)



We are going to take about small problem, which is finding the largest of three numbers, it is a fairly simple problem. But, in this problem by looking at this we will see, how we can take a problem break it down into a smaller chunks, we will see a how to do a flow chart or at least come up with the sequence with which you can see solve the problem. And I will also show the rudiments of how to write a C program and the software environment that is required for it. So, let us jump write into the problem.

(Refer Slide Time: 00:44)



So, we are looking at largest of 3 integers or 3 numbers. So, let us call them A, B and C. So, this forms the input to the program or input to the problem and what we are interested in finding out is, what is the largest of the 3, is it A? Is it B? Or is it C? So, we have to start some variant, we are start with comparisons. So, let say I compare A and B I am going to check, if A is greater than B or not. There are two possibilities A is actually greater than B or it is not.

So, if A is greater than B then it still possible that C could be greater than A. So, we go and check if A is greater than C also if it is yes, then indeed A is the biggest of all. So, you already checked it against B and you have checked it again C and therefore, A is the largest. However, if this condition is not true, if A is actually less than C, we know that A is greater than B therefore, C must be the greatest of all. So, this A is greater than B, but it is less than C therefore, C is the largest.

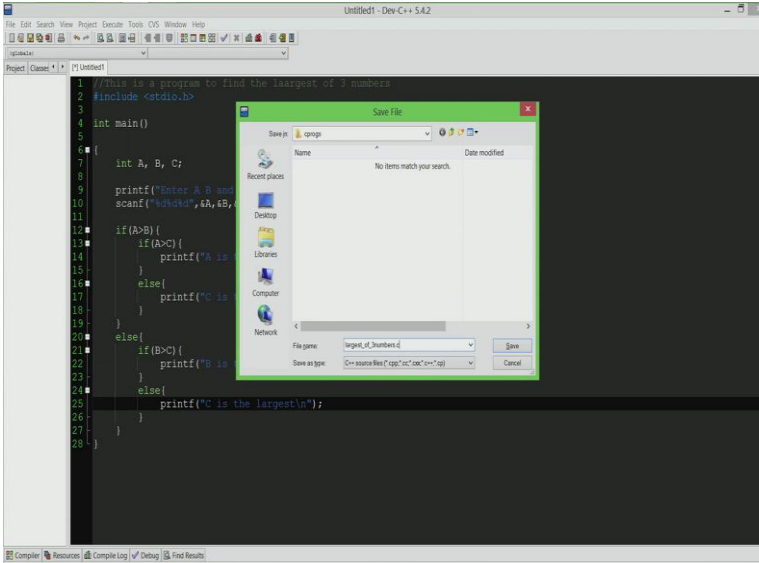
Now, let us look at this other side A is greater than B is false, which means B is the largest among A and B. And now we can also check, if B is indeed the largest by check comparing it again C, if B is greater than C, if it is true, then B is the largest; otherwise, C is the largest. So, let us take one look at this problem, before we go and write a program for this, we first compared A against B and if it is true, we checked A against C, if both are true, then A is indeed the largest. If A is not greater than B, then we checked B against C. So, if B is greater than both A and B C.

So, if B is greater than A, but if it is less than C, then C is the largest, if B is greater than both A and B then B is the largest and so, on. So, this is the basic setup that we have. So, at this point of time, it may make sense to try with a few example and check this is indeed correct, if this solution is correct. So, let us take this example A is 1, B is 2 and C is 3. So, let us check this is A greater than B is the first thing that is checked. So, you compare A and B is A greater than B no, A greater than B the answer is no.

Now, you are ready to check is B greater than C. So, to compare B and C, we take B, we take C is B greater than C. So, B is not greater than C. So, the answer is again no and therefore, the result must be C. So, essentially you have taken this path, let us take another example A is minus 1, B is 5 and C is 2. So, again if we check A greater than B it is false and B greater than C is true. Therefore, you have taken this path B is indeed the largest.

So, with these two examples I am now fairly convinced that this solution is correct. Let us go and write a program for this and you are going to write small c program and in the process I am going to show you, what it means to write a program. So, I am going to use this environment or an IDE also called an Integrated Development Environment called Dev C plus plus. It is a piece of software that is available for windows and it is available for free, it is widely used, lots of downloads, I can provides you links later for how to download it and how to run it, I already have it installed.

(Refer Slide Time: 05:26)



```
1 //Write a program to find the largest of 3 numbers
2 #include <stdio.h>
3
4 int main()
5 {
6
7     int A, B, C;
8
9     printf("Enter A B and C\n");
10    scanf("%d%d%d", &A, &B, &C);
11
12    if (A > B)
13    {
14        if (A > C)
15        {
16            printf("%d is the largest\n", A);
17        }
18    }
19    else if (B > C)
20    {
21        printf("%d is the largest\n", B);
22    }
23    else if (C > A)
24    {
25        printf("%d is the largest\n", C);
26    }
27 }
28 }
```

The screenshot shows the Dev C++ IDE with a C program for finding the largest of three numbers. A 'Save File' dialog box is open, showing the file name 'largest_of_3numbers.c' and the save location 'C:\source files\cpp\dev_c++\'. The dialog box also shows 'Save as type' set to 'C++ source files (*.cpp;*.c;*.cc;*.c++)'.

So, let me start a new program. So, I am going to start a new program and I am going to write a piece of comment in it saying, this is a program to find the largest of 3 numbers. So, I will be able to take input from the user, I mentioned earlier that we need a library for input and output and that is going to come from `stdio dot h` and I am going to write the main program. So, do not worry about the mechanics right now.

So, as I mentioned earlier main program is the very first instruction, that is executed and to do that we... So, the body of the main program is enclosed within braces. So, I am going to have 3 integers A, B and C, I am going to prompt the user for the values. So, `printf` is a statement which will print things on the screen. So, it will print enter A, B, C on the screen at this point of time we expect the user to actually print these numbers.

And there is some scanning that we have to do or reading from the user and we do that with this library function called `scanf`. So, do not get scared by the statement that is there. So, all it is doing is reading A, B and C. So, and you can see that every statement ends with the semicolon so, that is the terminator. Now, let us write this program. So, we want to be able to compare A against B and A against C and so, on. At this point of time, it may make sense to actually go and look at the diagram once.

So, we first compared A against B and based on yes or no we did further comparisons. So, let us compare A against B first, if A greater than B, then A still might B less than C. So, we have to check if A is greater than C also and if it is, then we are going to declare A as the largest. However, if this condition fails, then C is the largest. So, again do not look at the mechanics right now. So, just follow the logic we already had this, let us go down further.

So, we have taken care of two cases, we have looked at A greater than B and we have seen, this side we have taken care of this side. Now, we are ready to write code for the right side. If B is... So, by this time we already know that B is greater than A, we want to check if B is greater than C also, if B is greater than C already we know that B is greater than A. Therefore, B is the largest; otherwise, C is the largest. So, this is actually a complete program.

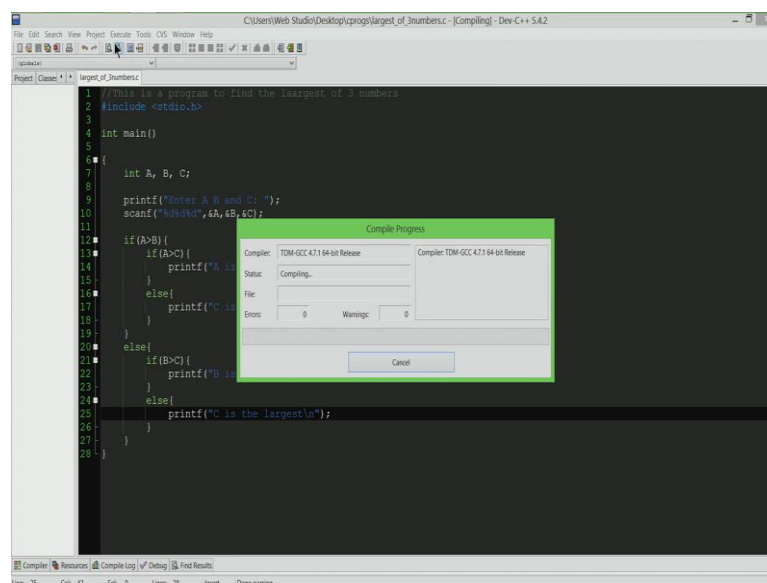
So, we have at line 2, So, you can see the line numbers on the left side at you can see it here. So, watch the mouse pointer at line number 2, we have `hash include stdio dot h` starting at line 4 ending at line 28 is our main program. And the body is from line 6 to 28, there seems to be something which is `int A comma B comma C`. So, seems to be

reading in the values of A, B and C here and we have some checks and we have some print statements. So, this is the whole process.

So, what we have really done is, we have prompted the user for entering A, B and C we read it from the user and we have written this piece of code, which actually declares whether which one of A, B or C is the greatest of three. So, let us save this program, at this point you are prompted for a name. So, I am going to write largest of 3 numbers. So, I am giving fairly a big name to the program, but you may want to give a smaller name if you wish to. So, I have this.

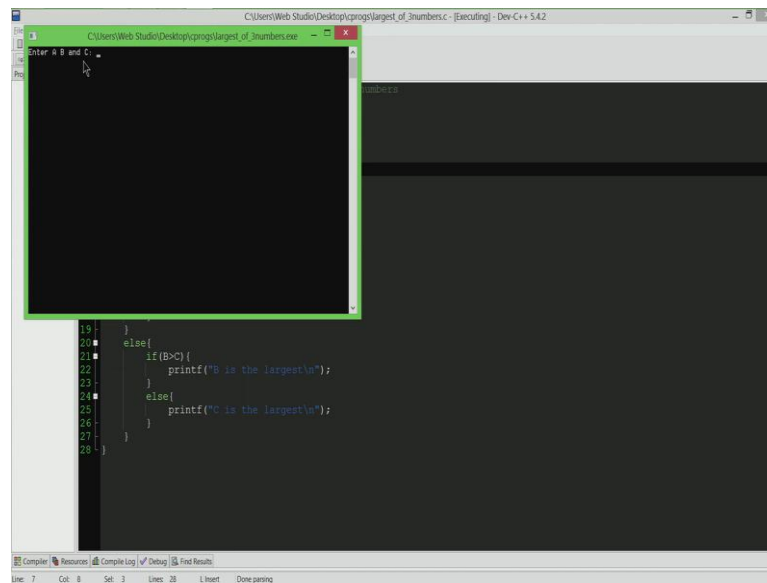
And this is not going to result in any thing. So, we just have a program by looking at the program one can usually check whether things are correct or not. But, the proof is in the putting, we have to a really make the computer run it, we have to give it test cases and see. So, we started with the specification we have the design which was the diagram that I did earlier and we have coded now. Now, let us go and compile it. To compile it you go to execute compile in dev C plus plus and while a we do not have any else.

(Refer Slide Time: 12:04)



So, have return a program which has no compilation errors, we are ready to run the program now, some I am going to press F10 for it, when press F10 the first statement that is there is printf enter A, B and C. So, this line number 7 is really not an executable statement, we will see why it is so, later, it is actually called a declaration. So, we have declared variables A, B and C of a type called int which transfer integer. So, the first executable statement is actually printf. So, printf enter A, B and C.

(Refer Slide Time: 12:52)

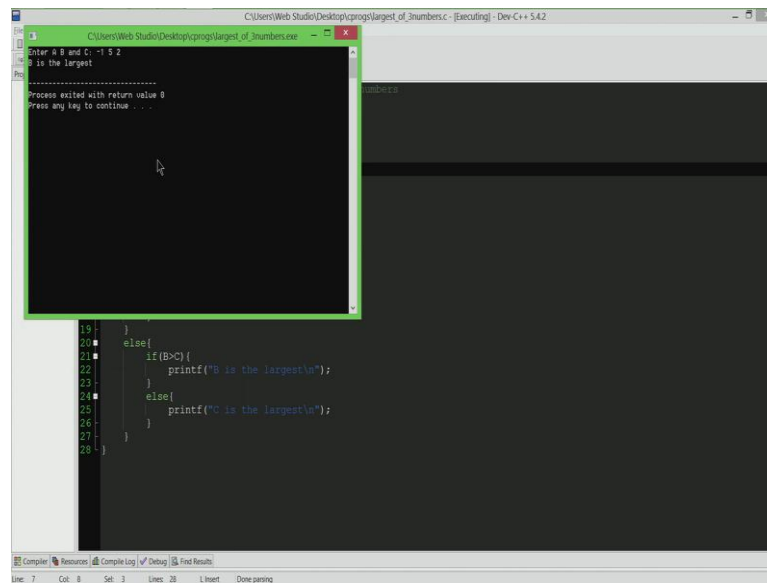


```
19 }
20 else{
21     if(B>C){
22         printf("B is the largest\n");
23     }
24     else{
25         printf("C is the largest\n");
26     }
27 }
28 }
```

And if you see this screen, it just shows enter A, B and C, you can see that the double quotation, the open and close code is actually missing in the output. So, this is the way in which printf which is a library function takes an argument. So, library functions are just like mathematical functions. So, if I ask you what sine of x is, you would write it as sine with in parentheses x would you not? It is same thing here. So, printf is a function with in parentheses, you pass the so, called parameter to it. And in this case the parameter is the string enter A, B and C and we can see it on the screen here. So, now we can see the blinking cursor here; that means, the user is expected to enter something. So, let us take the two examples that we gave earlier. So, let us go back to the example ((Refer Time: 13:44)) we tested 1, 2 and 3 and minus 1, 5 and 2. So, I am going to enter 1, 2 and 3 I am type the key from the key board and it says C is indeed the largest. So, far, so, good.

Let me go back and try another input. to do that I am not really change the program per se, which means nothing in the logic of the program has changed, I could just use the executable directly this what I mentioned earlier, you could write it once, compile it once and run it how many ever times you want. So, I am running it the second time without making any edits to the program, because I do not see any evidence that the program is is erroneous.

(Refer Slide Time: 14:29)



```
19: }
20: else{
21:     if(B>C){
22:         printf("B is the largest\n");
23:     }
24:     else{
25:         printf("C is the largest\n");
26:     }
27: }
28: }
```

So, enter A, B and C. So, I am going to use minus 1, 5, 2 as an input and it says B is the largest. So, we have tested two cases this does not mean that it would work for every case. So, what I am going to do is, I am going to just try and give all possible combinations of starting from ((Refer Time: 14:52)) let us say 1 2 3 I am going to try these combinations 1 2 3, 1 3 2, 2 1 3, 2 3 1, 3 1 2, and 3 2 1. So, I am going to try these four and these two combinations, one after the other and if there is any bug I can hopefully catch it. So, let me give it one option after the other.

So, I am going to run it once 1 2 3. So, C is the largest good, then let me run it once more 1 3 2, B is the largest good. So, I am going to try 2 1 3, C is the largest, 2 3 1, B is the largest, 3 1 2, A is the largest, 3 2 1, A is the largest. So, I give six permutations that can come from ordering 1 2 and 3 and it looks like this program is actually correct. So, this by itself this not show that the program is correct for the input that I given is correct and it this case it would actually sort any number.

So, we have also tried it on negative numbers. So, let us try all negative numbers it say minus 1, minus 2, minus 3 and in this case A is the largest. So, just for fun let us try instead of giving numbers, let us try something else. So, I am going to give input as let us say S t and R. So, I gave three letters and I did not give numbers and it actually says B is the largest. So, what happened here? So, we gave three letters S t and R and it says t is the largest, for whatever reason. We will see why, how this happened? We will see why the B is declared as largest. So, what is in fact, B what is, A and what is C, we will see in a later module.

But, as of now if I give numbers things seem to work. So, just for one last time I will try another input 200, 100 and 1000 and C is the largest, so, far, so, good. So, this we started with the problem and we went through a sequence of steps and we designed a solution to the problem and the design is here. So, this diagram is the design and once we have the design we went and coded it, we check the code by testing it. So, we have the whole setup.

So, let me close this IDE or Dev C plus plus. So, with this we are at the end of this demo, we saw this very first program that we wrote, we tested it, we compiled it, we tested it and we ran it multiple times to ensure that things work out.