Lecture Transcripts
Motivation for
Data Structures and Algorithms

Welcome to this session, in which we will look at some more examples of 'Data Structures and Algorithms' and the motivation for studying these.

So let's look at another example program which is slightly more complicated than what we saw the last time. So this program reads a triangle and outputs its area, so we'll come back to this program later, but let's see what is needed for doing this. So this program needs several data structures, there is no built-in data structure for triangles. So unlike an integer value which you can directly read in, there are no functions available for reading a triangle and there are no variables that you can declare that say that this is a triangle. But C++ gives us ways of defining our own data structures for different objects and there is a way of representing triangles in terms of points. So, a triangle is nothing but a set of tree points in the plane and a point in the plane can be represented by its x and y coordinates and the coordinates themselves can be represented by a built-in data structure called a float, which is used to represent real numbers. So let's go back and see the declarations. So we have declared a structure called point, which stores two floating point values.

So this will be used to represent a point in the plane and a triangle there is a set of three points in the plane. So we are using, declaring a type called triangle which stores in an array with three points. So we built more complicated data types using the built-in types and using constructs in C++ that allow you to define other types.

So now, these are some of the functions, so unlike integers where you have direct ways of reading in and writing them, you don't have functions for reading in points or reading in triangles. So this is a function which just reads in a point, which just takes the two coordinates of the points and assigns it to the corresponding fields of the point variable. Now similarly, we need a function for reading a triangle, because there is no built-in such function. So a triangle is a set of three points, so we will call the function for reading a point three times inside this triangle. It reads each individual point of the triangle and that in term reads the two coordinates of the point. And now, this is the area function for this, of course, we need to define how to compute the area of a triangle, so from your geometry classes you would know how to compute some formulas for computing the area.

So this just uses the coordinates of the three points of the triangle, corners of the triangle and computes the area and returns. Now, we've main functions simply reads, calls the read_triangle function, compute calls the area function for that triangle and outputs the area.

So let's see this program again, there are several data structures used here. As I said there no built-in the data structure for triangles. So we define the type called triangle that in turn requires us to define a type called point and a point itself is represented by two coordinates and the coordinates can be represented by a built-in type called float, which represents real numbers. So, we are building more and more complicated types from what is available and the more complex programs you write the more complicated such structures you will need to build-up. So this algorithm also uses four functions. So you need a function to read a point and another to read a triangle because they are not the built-in types, you need be able to read and train them.

In this case we didn't need to print a point or print a triangle, we didn't write any function for printing. But in general per types, you will need to write function for printing them also. Then you need a function to compute the area of the triangle, of course, you need to find how to compute the area, we need to figure out what formula to use, that is something independent of the program, once you do that you can write a program to compute the area. Then the main function simply reads the triangles and outputs its area, but the main point to note is, we don't have built-in functions for these, and you need to build your own data structures and algorithms for reading in these types and manipulating them.

So in general, in a programming problem you need to identify the data structures and algorithms that are needed to solve the given problem. In this case, the data structure needed was a triangle and algorithm needed was how to compute the area of a triangle, but for this you may need other data structure, so there is no direct way of representing a triangle. A triangle is a set of points, so you again need data structure points per points and you may need other algorithms for input-output and so on. So for reading and writing triangles and points.

So we need to define our own types of variables to implement required data structures, we need to define our own functions to implement the required algorithms, and there are many different implementations possible for the same data structure or algorithm. So there may be other ways of representing triangles or other ways of representing points, for examples instead of "xy" coordinates you can use "r" theta representation, where you will represent the distance from the origin and the angle. So there are many alternatives available and you need to choose the correct alternative for your particular problem. So here are some exercises for you, so try to find an alternative data structure for triangles.

For example, instead of representing all the three corners you could think of representing the sides of the triangle and then try to compute area based on the lengths of the sides of the triangle. So you could use a different formula for the area of the triangle. So there are many ways in which you can write the same program, and you can now write many other operations on triangles also. So once you have a data structure for representing triangles, you can write many other functions for performing operations on triangles. So one exercise is, try to write a function which takes two triangles as input. So you can use the read triangle function for reading in the two triangles. So previously, we had a function for just adding two numbers, now we take two triangles and determine if they have a point in common. So, it outputs yes or no if the two triangles intersect or they don't intersect. So you can use the same data structure that we have used in this example program, but you will need to write your own algorithm for deciding, how to check whether two triangles intersect, two triangles specified by their corners whether they intersect or not. So try writing this as an exercise and will continue later.

Thank you.