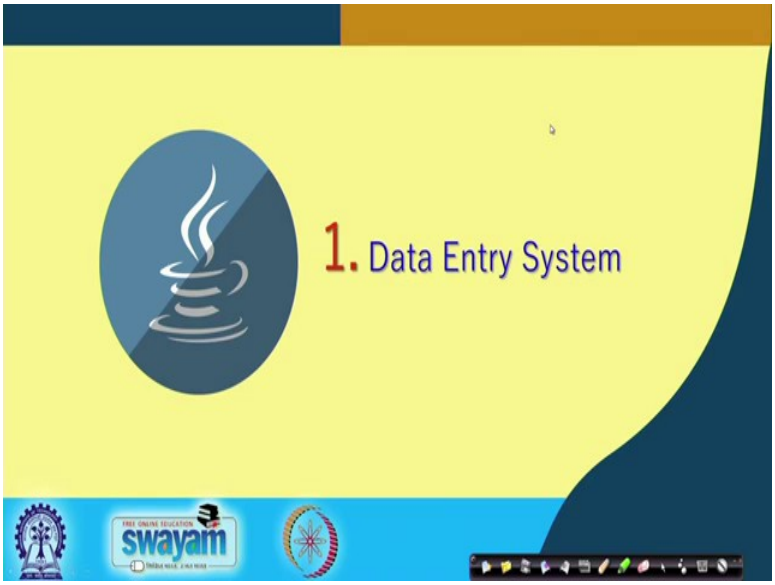
Programming in Java
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 56
Case Study – I

So, this is the last week of this course, it is the 12th week. Now, so far we have learned a lot of regarding the Java Programming. And it is our now time to exercise, so practice matters. In any programming, language learning, you can improve your skill only if you can indulge yourself into doing some work, and then practice and practice.

So, considering these things, we have planned few projects that you can take off, so that you can deal the projects, try to implement, and then ok, then you can learn many more things, which basically is not possible to learn, only just using or attending a course through video course or whatever it is there. So, a project is very important. So, now we will discuss in this week, a bunch of projects will be discussed, how to solve all these projects. And to solve these projects, what are the other skillset that you should have, all these things will be discussed one by one.

(Refer Slide Time: 01:33)



So, let us have the first projects that we are going to discuss about. It is a data entry system. So, by the name you can understand that, we have to develop one software by which a data can be entered. Now, obviously data entry should be related to a certain

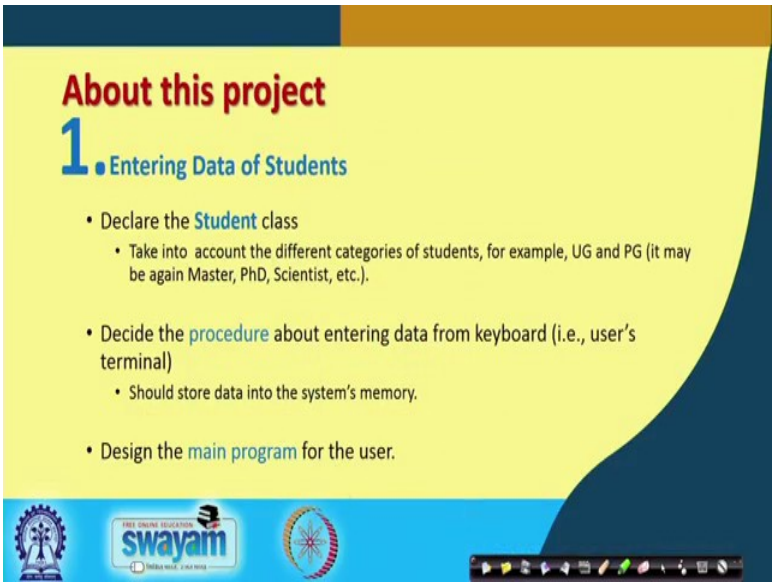
application. So, here we will consider the application regarding the data entry for the student's record.

(Refer Slide Time: 02:01)

So, basically for this project, you need basic concepts about encapsulation, inheritance, and then information hiding in the form of access specifier, the concept of package, and little bit GUI programming. As you can see, we have to develop one program, it is just a mini-software we can say. If you can develop these programs, then at a later stage any large scale program also can be developed, it is similar to the same application.

Now, data entry application is huge, it is an everywhere actually. And this program is a simple one that we have to plan, so that you can do it, maximum time that you can think for solving this project only 1 month. So, in a month, in fact, for some students, it will take only 1 week in fact. Anyway, so at the maximum, you can spend time to implement in a month only.

(Refer Slide Time: 03:11)



Now, let us discuss about the idea of the projects. Here the idea is you have to declare a class. The name of the class, you can give as a student. This is because we are going to enter the record or information about a student, so there should be a class declaration for that.

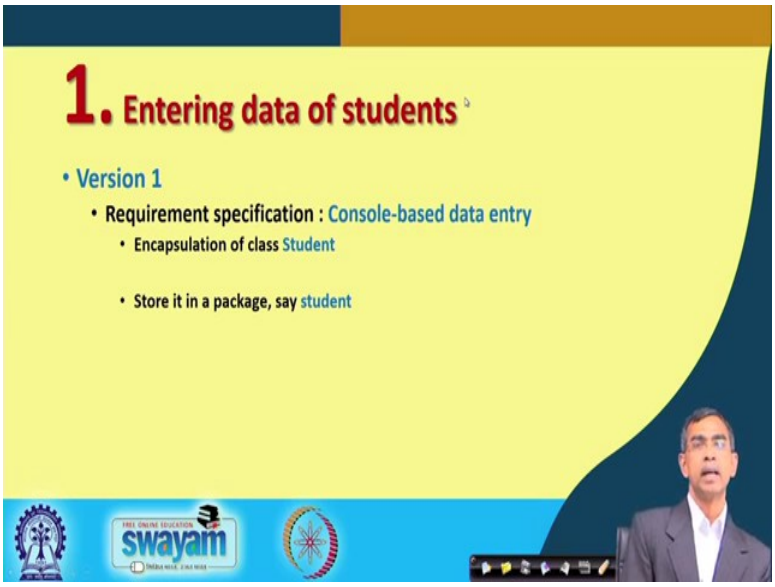
Now, this class should take into account, the different varieties of students. Students may not be the uniform only; there may be different category of students. Like some students may be studying only undergraduate course, some students may be postgraduate courses, namely M.Tech or Ph.D., there may be some higher than the PG course also, say post-doctorate fellow or scientists like. So, we have and obviously as you know, so data entry for UG students may not be the same the data entry for a PG student like a research scholar. This is because there may be some methods which are required which does not required in case of another category of objects.

Then we have to decide a procedure about data entering from the keyboard, so it is first we will consider about console-based data entry from the key keyboard, how a user can enter the keyboard, it will give a prompt that enters the name of the students. Then from the keyboard, you will give the data, and then return. Then again it will prompt for what is our roll number, you have to enter this kind. So, this is the console-based test entry. Usually, you are familiar with this kind of entry, when you are learning about C programming or something like that. Anyway, so this is a very trivial job.

And then finally, we have to do the same idea using some different way. A different way in because whenever, you use the console-based test entry; the data usually stored in your main memory and so long your program is in execution, the data will be available. However, if you quit the program or your system is goes off, then your data will be lost. On the other hand, the data that you entered even if it is a console-based also, you can pump it into secondary storage, so that in a file the data can be stored.

So, here data entry also very important that how you enter the data, and where this data ultimately goes to, so it is basically a file. And then finally, we will consider about a little bit sophisticated data entity, it is not console-based. It is basically, we can develop a GUI windows program which allows a user to enter the data prompt will be in the GUI itself. And then data once it is entered from the GUI, it will directly go to the file. Anyway, so this is the idea about this project.

(Refer Slide Time: 06:09)



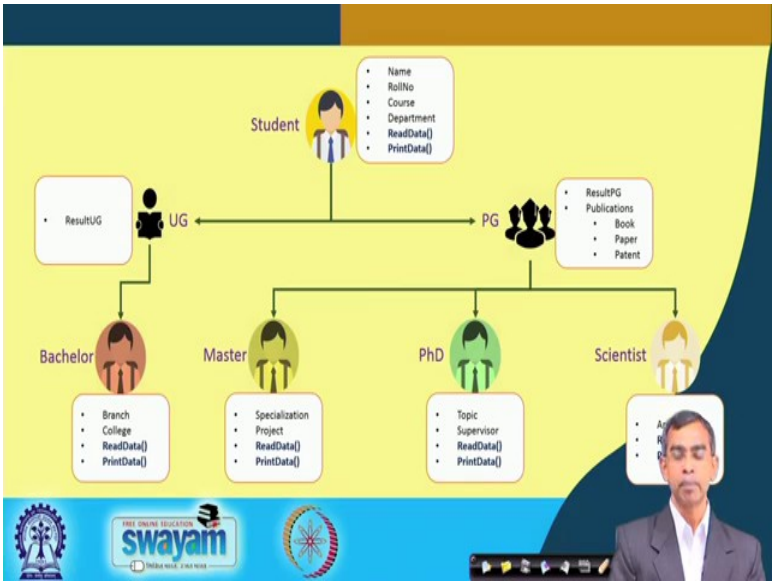
Now, let us have different steps. So, it is better than if the project is of large size, we have to solve this project in a facet manner. I am giving an idea about how in phase by phase, this project can be implemented. So, the first version of the project that we have planned here that console-based data entry. And for these things, we have to have the class declaration.

So, encapsulation of all the class two of the students based on inheritance also. And all the class that you will build based on this class declaration, they should be stored in a

package. Because, whenever the large classes are involved; the better idea that you should not store all the classes in on directory, instead of you could you should maintain all the classes in a several directory in a different directory.

This will basically help to manage the program at a later stage. It is basically required for either modification or up-gradation to a latter version like, so that is the package concept that you should exercise from here. So, all the class declaration that you will give, it is basically in the form of a package. I am giving you a quick look at that how a class of the students will look like.

(Refer Slide Time: 07:41)



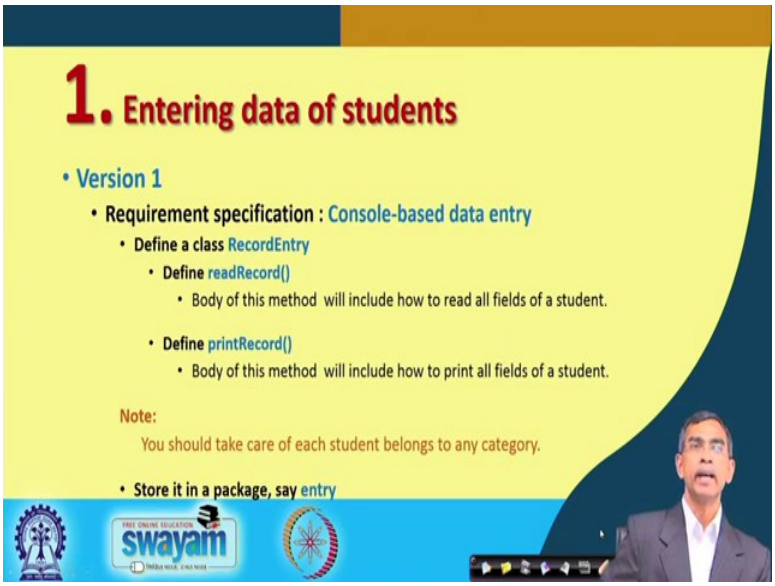
So first, you have to decide a base class, the name of the base class that you can give as a student. And then student base class can includes the fields and methods which I have mentioned here as we see here. So, these are the different methods and fields not limited to this one, you can think something else also that may be more suitable. I have given here for your just a hint.

Now, so this is the base class. And then as you have told you, the class may be different, then all students belong to this class student actually. And then there will be a category is that UG, so UG has again different fields which are not there or there may be some method, which needs to be overwritten there or the method you can declare here, you can reuse to define these method here. So, this is the concept of inheritance that you can follow is a single inheritance Java provides only, you use this one.

And also some other data that you can include which is not here, so result UG. And this result UG itself is another class ok. So, a course may be another class, because a course can include some other, which is called the subjects also. Course under this course subject 1, subject 2, subject 3, and for each subject again, they are maybe marked 1, marks 2, marks 3. So, this basically is a composite subject.

Now, it is up to you up to which details level that you want to go. So, it is basically an example of a nested class that you can do, even the class also. All those classes can be declared in the package, and I can use it by means of import and others. So, this way the bachelor class can be declared, like master class can be declared, Ph.D. and scientist class can be declared. And they are a different class because it has certain fields or methods which is not there either here or there, so that is why the different class compositions are to be taken into account ok. So, this is about the different class concepts that you should follow, and then store them in the package as we have mentioned that student package it is.

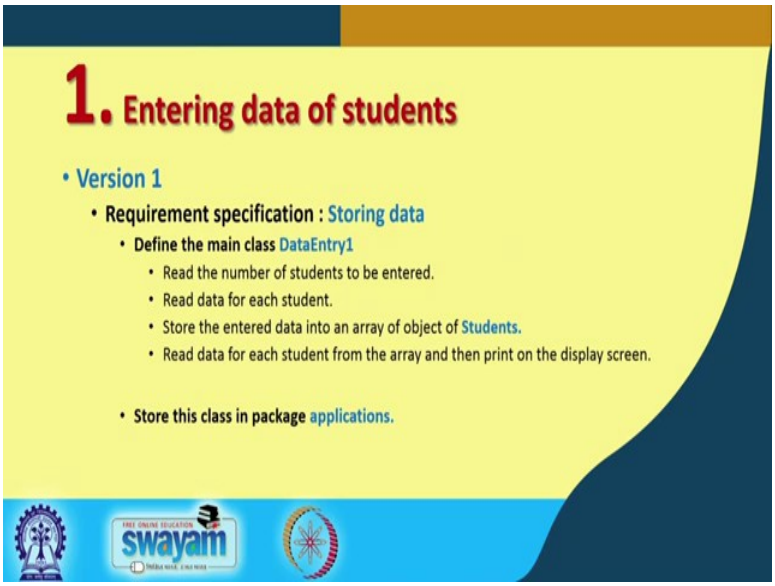
(Refer Slide Time: 09:51)



And then our next step that you should after declaring the class and everything, you have to create the objects. There may be n number of students can be created belongs to a particular category, you can create the array of objects. And for each array of objects, you can read the field values, the member elements from the keyboard, so that is what the idea about.

And then once you read it, whether you are reading is successful or not, then you can just define a method the printRecord, so that all these things the data entry through the console, again you can plan as a separate class. And all these classes can be put into another directory let the name of this directory that is the package as an entry.

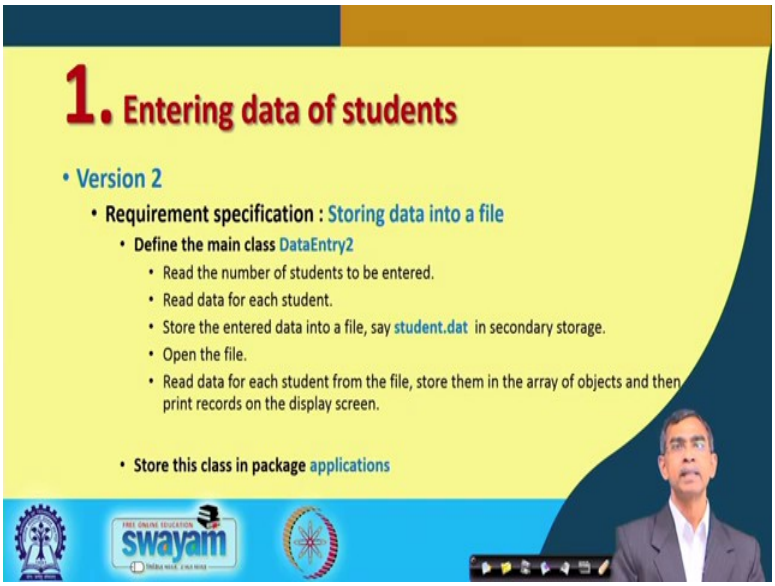
(Refer Slide Time: 10:37)



So, this is the second part of this first phase, namely entering the data. And then here is the next phase of this is storing the data. As you have seen that data that you have stored, it is stored in the objects only. And those objects are stored in an array, and this array is basically in primary memory.

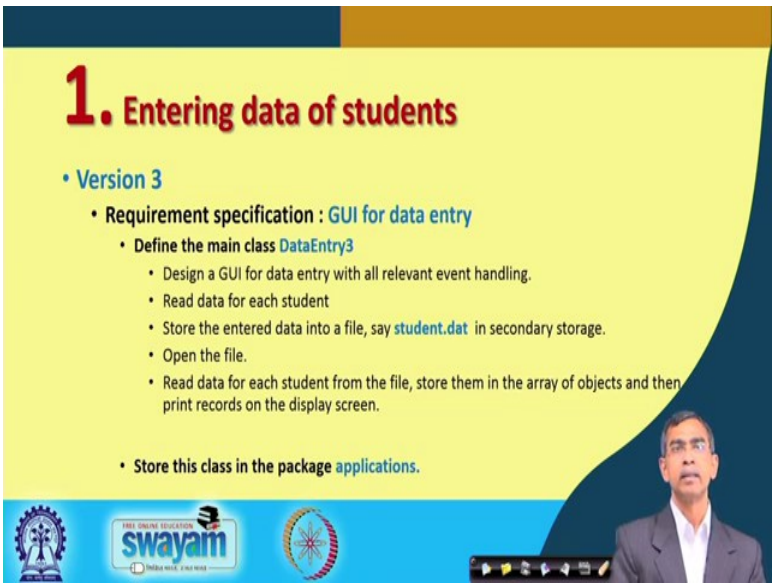
Now, so you have to store all the objects into a file that means the values that is there in the different fields of objects should be passed into file. So, in this case, you can use the IO stream handling more basically, you can file handling the class, and then use it, and then you can do that. And then you can save this class for the data entry in another package application. And this will be your main class actually. The data entered in this case is the main class. So, in the main class, you can use all other class that you have declared in the different packages like students, and then data the entry package it is there. So, this is version 1.

(Refer Slide Time: 11:49)



Our next version is basically; so long we have entered the data, and then store it in a file. Our next version is basically data entry 2, so we can store earlier ok. In the next last version, you can think about that do not store the data into any file. Now, in the next version, we can store the data into a file. Everything is same only the thing is that the input from the may primary memory will go to the file memory actually.

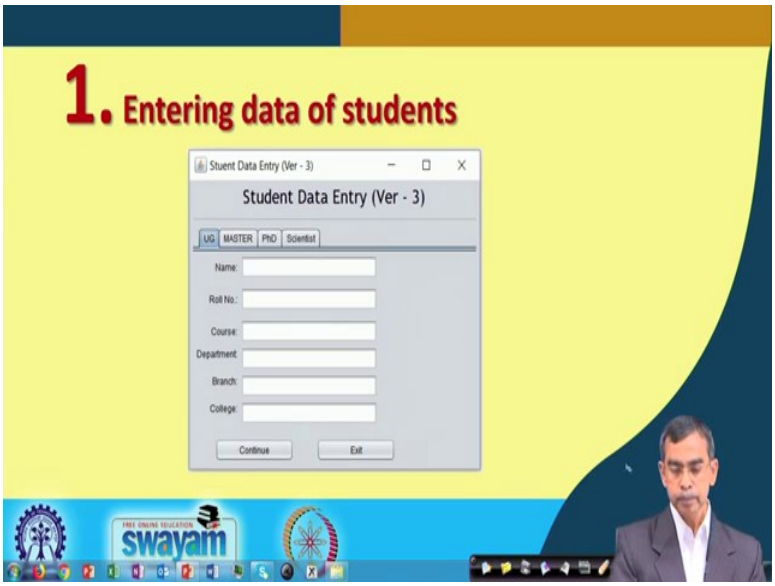
(Refer Slide Time: 12:13)



And then this is the version 3, the third phase of the program we can say, here you have to define a GUI for data entry. Now, here again, the data will be entered from the

graphical user interface that means some text field, and then some buttons will be there. All those button also needs to be handled properly and then keyboard handling event should also be implemented. And then once the data is entered one by one, it will basically go to the file again.

(Refer Slide Time: 12:47)



Now, I am giving a quick lookup, how a data entry skill would look like, it is a very simple one, you can use the AWT package for these to develop it. And in this case, you have to have a this is basically the frame, this frame contents level, these are the level. And these are a text field, and this is the button, and these are another button-like.

So, these are the basic components in addition to there is also some tab. So, if you want to enter the data for the UG student if you click it, then all the this basically the composition will change according to the data, those are relevant to the UG. Similarly, if you end tab this one, then this text field actually will change according to the different field which is relevant to this data.

So, this way you can have the tapped text entry system, and all this can be done using even java swing also, AWT also. If you use java swing, absolutely this is very good; otherwise, you can use also AWT. Anyway, so this is the way about the data entry that can be done. And a simple program that you can think about, and you can take this as a project. So, this is about the first project.

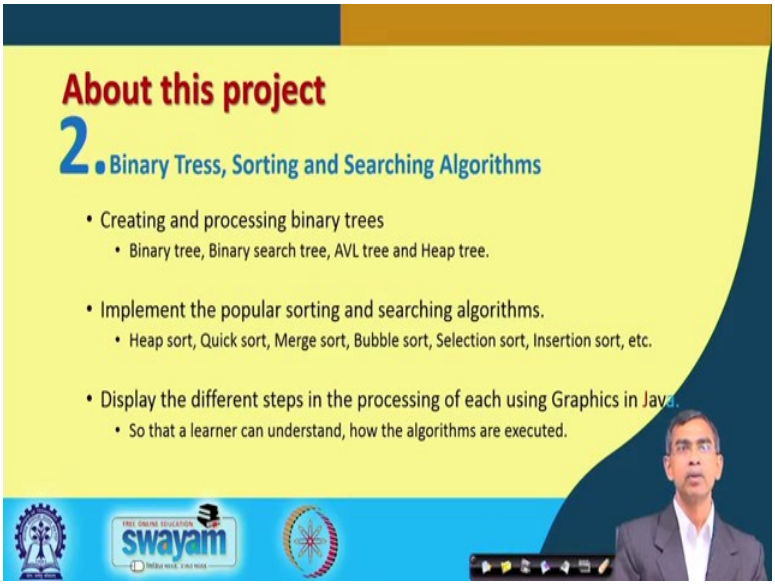
(Refer Slide Time: 14:01)

Now, I am coming to the second project. Now, those are the students, who are having some computer science background. This project mostly seems to be I mean feed for them only, because it needs some computer science background actually. But, other people also can do it, but in that case they have to learn some data structure related and algorithm related concepts. So, it basically this program is a computationally bit intensive, those are basically fond of solving problem, it is basically good for them.

(Refer Slide Time: 14:39)

Now, let us have the idea about. Now, before going to the idea, I just see, I just mentioned that what is the skill set or the programming confidence that you should have core programming that means, all the control structure language reference and everything should be there. Bit encapsulation, inheritance also can be a used here, package concept is there, polymorphism, scope rule, and then finally also the graphics should be here so that you can exercise graphics. But, here actually a little bit about AWT only, and not so much details graphics, it is there.

(Refer Slide Time: 15:15)



Now, let us see what is the idea of this project. Now, this project is basically all about to create binary trees, then sorting algorithms, and searching algorithms. So, the binary tree is concerned as there are many binary trees, but we can limit it to only a few that you can do. Even if you can do only one also that is good, not necessary to solve all the binary trees in fact, but it is advisable that you should try to hands-on all the binary trees actually. So, this will basically give a lot of programming grips, how it can be there.

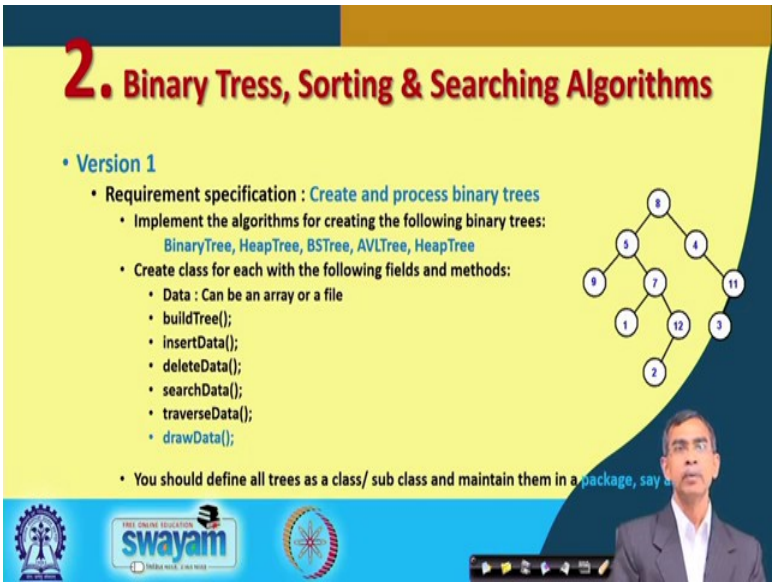
So, here we have mentioned binary tree, binary search tree, AVL tree, and heap tree. AVL tree is also called the height balance tree. So, any good book on data structures that you can follow, and you can have the information from there. And before taking up this project, you should have a good familiarity with all these concept trees algorithms, sorting algorithms for searching like this.

Now, so for the sorting algorithm is concerned, I have mentioned a few popular sorting algorithms which are basically easy to code actually. So, heap sort, quick sort, merge sort, bubble sort, selection sort, insertion sort etcetera. And then all the sorting and searching algorithms that you have developed, we will go for visualization. Even the binary tree construction also, I will come to the concept of visualization later on that means, each process of the algorithm can be visualized, so that anybody who does not know about how exactly the sorting takes place, they can see the visual progress. And then understand about the technique like. So, this is about the idea.

And here one thing you should note that java does not support pointer concept. And you may think that binary tree and everything how without a pointer, you can implement. Absolutely pointer is not required there, because you can maintain the elements that are required to be used in a tree can be stored in the form of an array.

So, array-based tree implementation is quite possible here, and you should do that. And for this array, you can use the data that package java.utit.package from there lot of other data structure, array, least, vector all these things you can use here. No need to defined or the array of your own which is in a primitive way, all the API based the least can be used for this purpose.

(Refer Slide Time: 17:37)



Now, the first version in the first phase that you should do is basically create and process binary trees. Here you have to implement the algorithm, those are there which are

available in the book of any book of data structures Binary Tree, Heap Tree, Binary Search Tree, AVL tree and that is binary search tree, and all these things ok.

And then whenever you declare you have to basically for each class, you should declare the different class in fact. For each tree, you should declare a different class in fact. And then this class should include the different fields as well as the different method as I have mentioned here, one field may be data. And data can be stored in the form of an array of integers if you want to build the tree for an integer data or if you want to build the data for a string, say an array of strings like this.

And then here the methods that you have to implement build tree, insert data, delete data, search data, traverse data, draw data. Draw data means basically for the visualization and others. So, all the classes that you will build, you can maintain them in a package. Here I have given the name of the package, I will go. And you can note that whatever the name of the package, you should give it should be in small letters. No capital letters, usually advisable to give the package name. This is because in many operating systems, the name of the directory is not case sensitive. So, capital letter, small letter, mixing the package name is not usually allowed.

So, you should give the only small letters like a, and preferably as a three to four characters max not more than that also. So, this is the package that you can develop. This package algo, we will include the implementation of all possible trees, implementation includes the different methods and fields, and also hierarchy you can mention, so that means, you can make the basic tree and then inherited into the different tree structure like a binary tree. A binary tree can have a special property binary search tree, heap tree is also another kind of property.

So, there may be an inheritance from the tree binary tree, there are one inheritance is a binary search tree. Another inheritance may be a heap tree. Then AVL tree is a special kind of binary search tree. So, AVL tree can be considered as a subclass of binary search tree-like. So, you can plan that how the hierarchy that means, encapsulation or inheritance tree can be considered so that with a minimum number of code or code reuse a using the method of code reuse or you can use it. So, this is about the second part first part of the project.

(Refer Slide Time: 20:27)

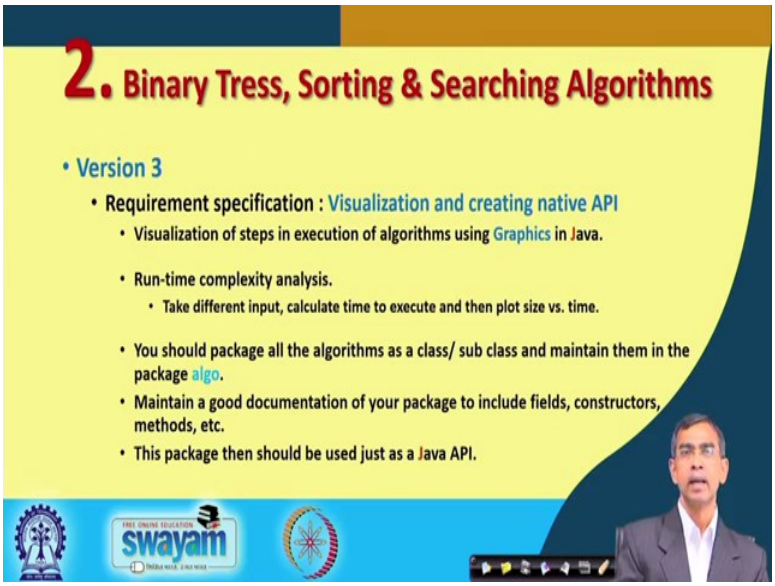
In our second part of the project, who have planned to implement the sorting and searching algorithms. The sorting algorithms, again it requires a lot of fields. So, for each sorting algorithm, you can plan one class. And this class will include the appropriate fields and the methods. And again like the tree also, you can maintain an inheritance structure, so that the different course that can be used, it can be store the in a general can be stored in the base class. The other subclass can have their own codes and own methods like this.

And then also the sorting searching algorithm can inherit that means, it can extend the package the class those are there in algo package also, I mean trees so because some trees also need to implement some sorting algorithms. For example, if you want to do the heap sort, so it is basically needed the heap tree. So, this heap sort extends heap tree, and then all the classes those are there can be utilized there.

So, if you can put into another package algo or in the same package also no issue, you can do that. And the methods, you can do some open, insert, remove, save that mean, after sorting the data can be saved. The opening means, you can open from the raw data input data, and then remove also. If you can eliminate one item from this data sorted an array or it can insert also, but insertion should be in a sorted manner. All those methods also, you can implement for each sort of techniques.

And also another method time is basically how much time that a sorting algorithm required to complete the sorting task or a searching task. So, the time also you can use as a method for a particular class. And all the implementation under this category can be put into the package algo, so this is the second phase of the project.

(Refer Slide Time: 22:27)



And the third phase of the project is visualization and creating native API. So, in this process, we will have a total visual impression of differently. For example, you are building a tree. When you insert one single node, it will give only one node. When you add another node, it will go to that tree, whether it is a left child or right child. So, accordingly the tree will be built. And then the building process can be visualized one by one as the things are going goes on. So, this is the visualization process.

And this process again to do these things, you should take the help of graphics package, this which is there in AWT java.awt.packaged only graphics class actually, which is defined in java.awt.package. And all the implementation related to the visualization, you can put into some other package or you can put in the same package also, because it is related to the sorting, searching, tree generation, and then visualization and everything.

Now, if you build one such package, it basically gives an idea about how a see a user can develops their own package, which can be treated just like java APIs like say java.util or java.io like this one. So, all these packages you can use, and then import also. And then for further development, suppose in some other application, you need to use all the

sorting and searching algorithms. So, import this package and use it. So, this way the code reusability can be enjoyed at the maximum level, and then you can use it.

Actually, the software engineer or any software practitioner, they follow this kind of process. They develop the code, code each stored or architech. And the archive code is used or reused for development, development or for the modification whatever it is there. And whenever you write the code, and then all the system and everything gives sufficient comments, so that all the comments at a later stage if any other successor to you, who can be given the responsibility to develop the program they can use it, so that comment is very much essential. Every line, every statement, every class should have a good comment and annotation, so that is will give you an idea about, how the two different projects can be developed. And I hope if you can involve yourself to implement all those projects, it will really improve your skill.

Thank you and best of luck.