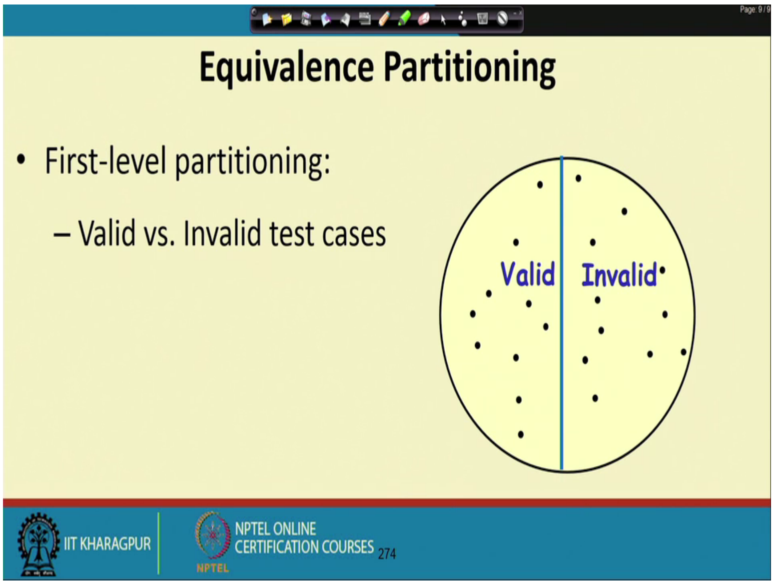
Software Engineering
Prof. Rajib Mall
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 47
Unit testing strategies-II

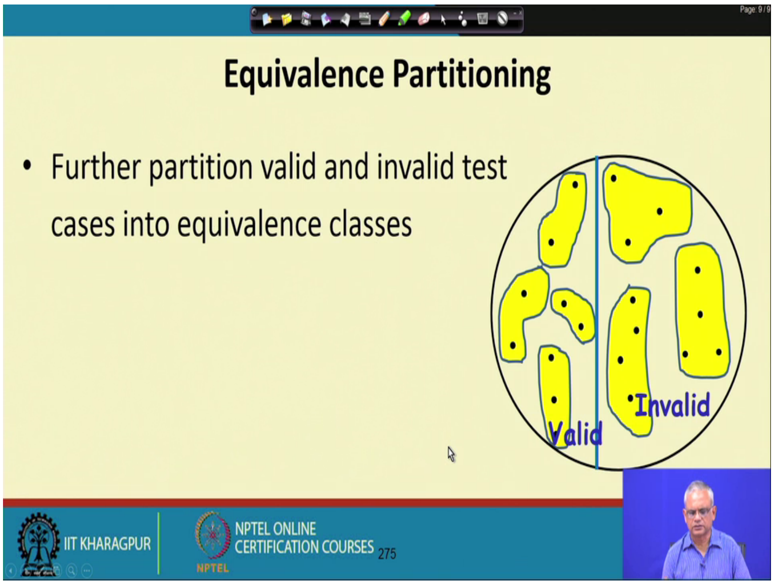
Welcome to this lecture. In the last lecture we were discussing about the equivalence class partitioning based Unit Testing. And, we had said that the equivalence class based unit testing. By observing the input and output data, we analyze the input and output data and determine the equivalence classes. And, once we identify the equivalence classes, we just take one data from each equivalence class and that forms our test suite. But, the main question that we are trying to address is that how do we go about identifying the equivalence classes.

(Refer Slide Time: 01:10)



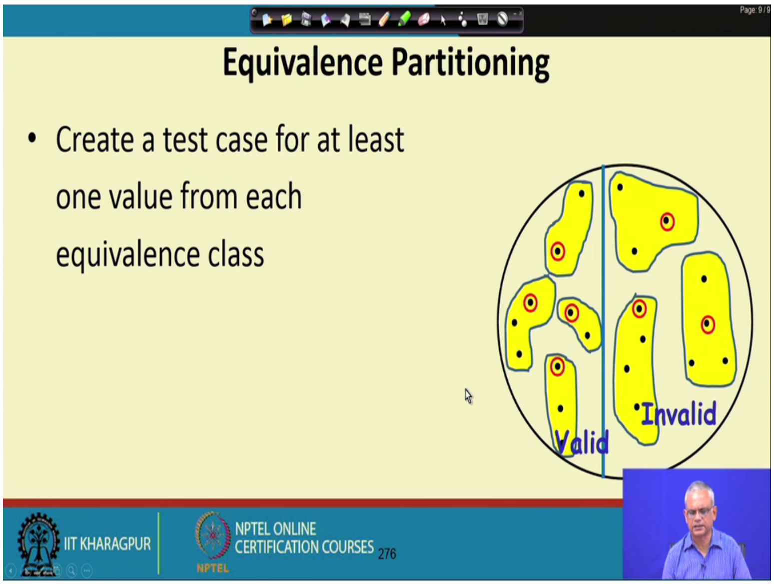
Now, let us continue our discussion further to identify the equivalence classes. We always can say that there are 2 types of equivalence classes; one are the valid set of equivalence classes and the other is invalid set. The first task is to identify what is the valid equivalence class? And, what are the invalid equivalence class? And once we identify these 2 equivalence classes, then we can identify further equivalence classes which are valid and further equivalence classes which are invalid.

(Refer Slide Time: 01:50)



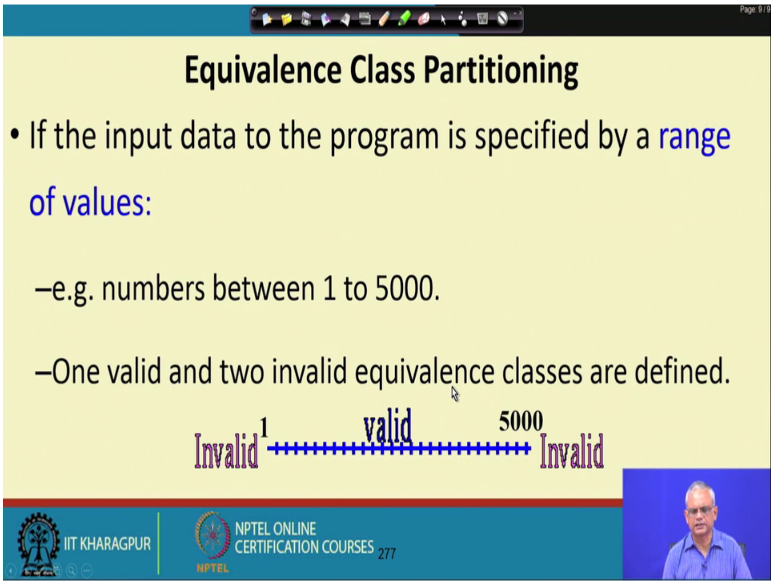
We have just represented that here that once we identify the valid and invalid equivalence classes, we can identify further valid equivalence classes multiple valid equivalence classes and multiple invalid equivalence classes.

(Refer Slide Time: 02:13)



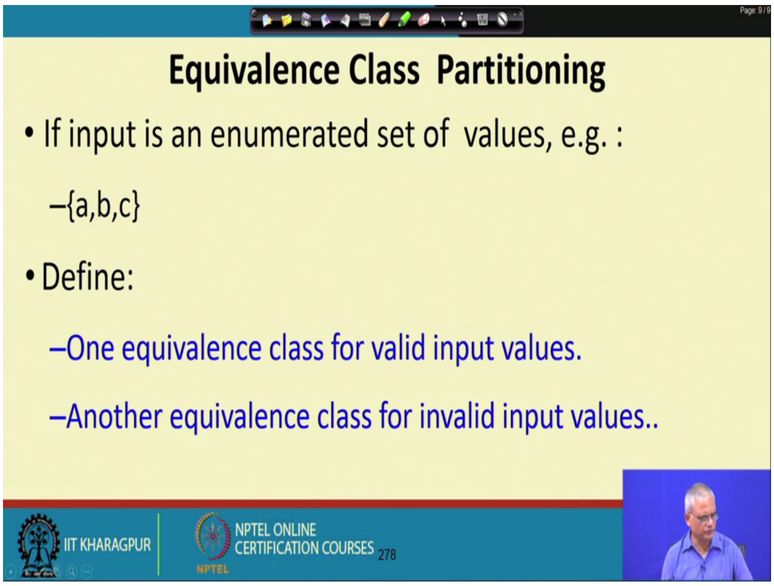
And, once we do that we can pick any value from each of this and that will form our test cases, but then the question is that how does one be sure that he has identified all the valid and all the invalid equivalence classes ok. It is more how to do with practice? Identifying equivalence classes is not really trivial that you just look at the problem and then say that see these are the valid and invalid, you need to analyze the problem. And based on the experience of solving many problems, we can say that which are the valid and invalid equivalence classes, but unless we have enough practice, it is likely that we might miss out some of the equivalence classes and therefore, the equivalence class testing may not be done very thoroughly.

(Refer Slide Time: 03:29)



Let us take start with some very simple examples.

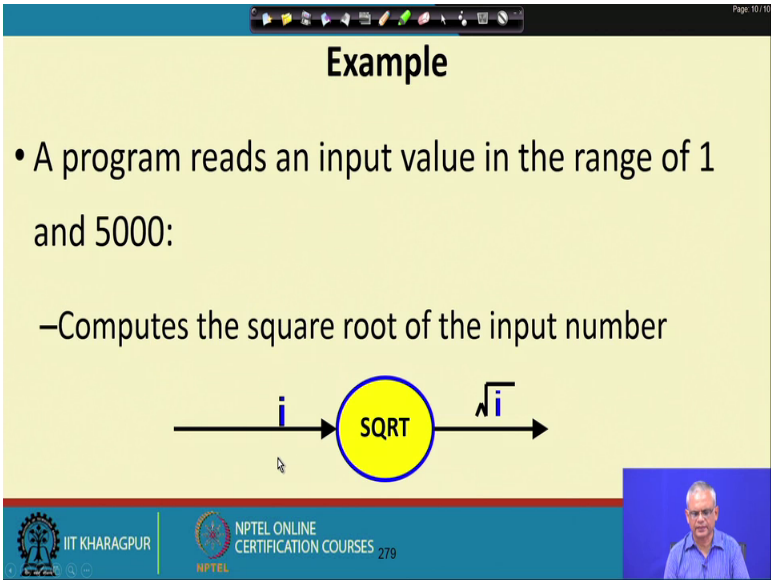
(Refer Slide Time: 03:35)



And see how to go about designing the equivalence class tests. The simplest example only 1 input data that is a integer number between 1 to 5000, an integer number between 1 to 5000. So, this is the range of values. And therefore, we have 1 valid equivalence class that is within 1 and 5000. And there are 2 invalid equivalence class 1 is greater than 5000 and the other is less than 1.

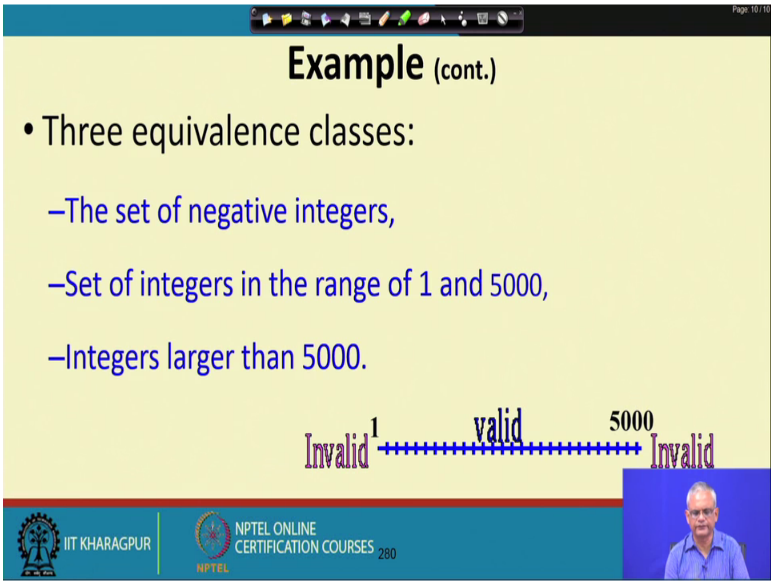
Now let us say a function takes a set of values let us say it only a b c these letters are valid and anything else is invalid. So, we have 2 equivalence classes, one is valid equivalence class, where we give a value a b or c and then there is another is invalid equivalence class, which gives a value other than a b c may be d or something.

(Refer Slide Time: 04:56)



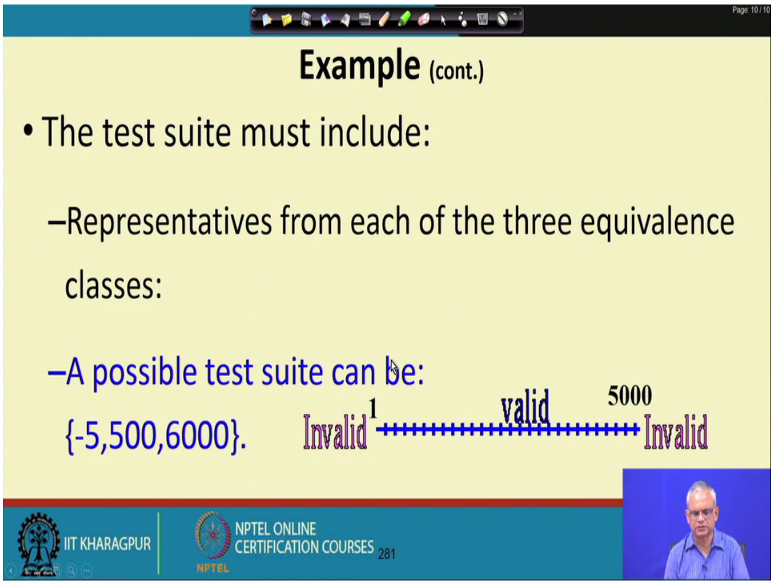
Let us take another example, where we have a function named as square root, which takes a value input value in the range 1 to 5000 and produces the square root. So, the input value by observing we find that there is 1 valid equivalence class and 2 invalid equivalence classes.

(Refer Slide Time: 05:23)



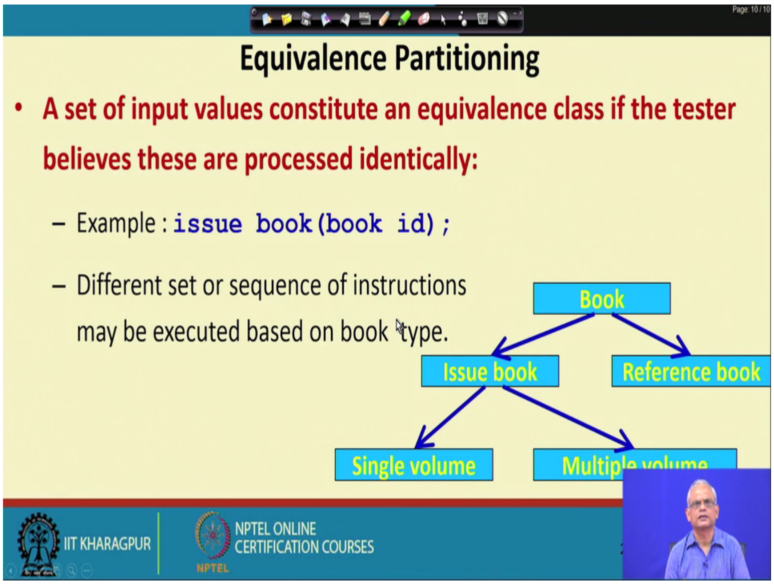
The valid equivalence class is anything between 1 to 5000, 1 invalid equivalence class is anything more than 5000 and then another invalid equivalence class is the set of negative numbers.

(Refer Slide Time: 05:41)



And, then we take 1 representative value from each equivalence class. So, 1 negative value, 1 value which is valid and 1 value which is more than the valid value which is 6 5000 is the largest valid value and included 6000. So, these 3 values form the equivalence class test for this simple problem.

(Refer Slide Time: 06:13)

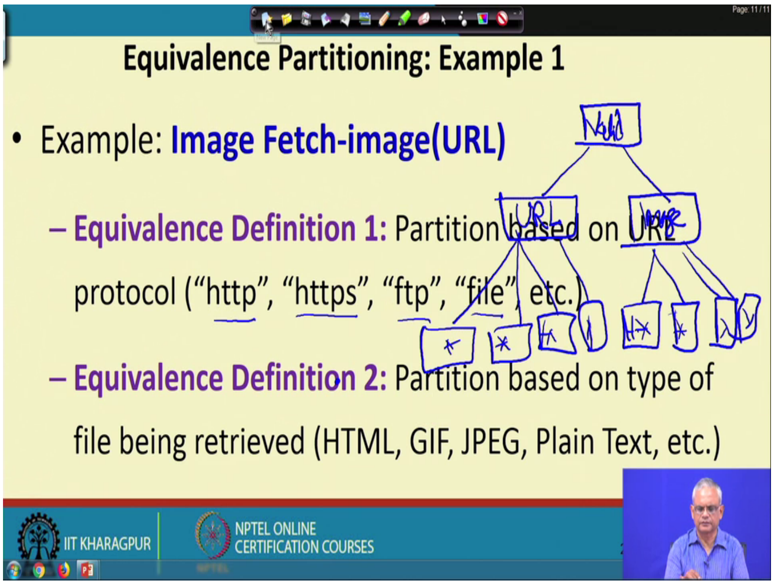


Now, let us look at another example. Let us say we have a function name of the function is issue book, issue book is the name of the function and it takes the id of a book may be the I S B N number or something and then it issues the book.

Can we identify the equivalence class for this function? We intuitively know how the issue book works just takes a book? And, then it issues the book the member of course, we have the valid and invalid set for this a valid book id and a invalid book id, but then among the valid book id, what are the equivalence classes? One is that it is a issuable book and it is issued out, if it is a reference book then the behavior is different will say that reference book cannot be issued out. Among the issuable books it may be a single volume book or a multiple volume book.

If, it is a single volume book may say it may just issue out, if it is a multiple volume book it may say that you have to take all the volumes together. So, the valid equivalence classes we might say that there are 3 of this. We need to pick 1 for single volume book 1 data item input data for a multiple volume book and another for reference book. So, for this example there are 3 valid equivalence classes, reference book, multiple volume, single volume, and there will be invalid class.

(Refer Slide Time: 08:50)



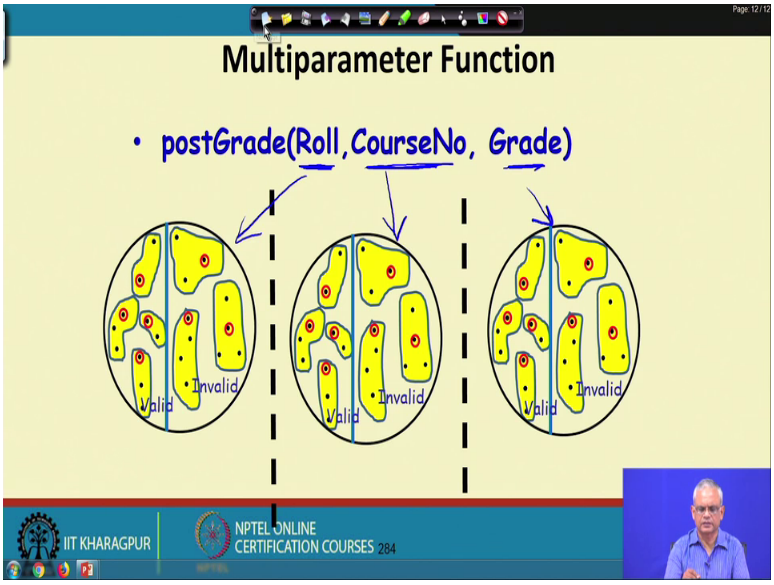
Now, let us look at another example. Let us say we have a function name of the function is fetch-image. We give a URL that is a website address and it the function fetches the image and returns the image. So, based on this simple description, what are the equivalence classes? One is of course, the valid set and the invalid set, but I am just asking, what are the set of valid equivalence classes? For this fetch image.

One equivalence class may be that, whether the URL is given in terms using "http" "https" "ftp" or it is a local "file". So, we need to these are the set of equivalence classes, based on how the URL is defined? Is the URL given for local file is a ftp https or http, but then for the same problem we might have another definition of equivalence classes.

That is what is the type of the image is it a HTML image, is it a G GIF image, is it a JPEG image, is it a Plain Text image. So, observe here that for the same one parameter, we have 2 equivalence classes that can be defined. So, for the valid set of equivalence classes, one is based on the URL protocol and the second is based on the file type and for the URL we have http https ftp and file.

And, based on the image type we have HTLM, GIF, JPEG, Plain Text etcetera. And, this forms our set of valid equivalence classes we need to pick one value from each of the equivalence class. The point that I want to emphasis here is that there may be different ways in which we might define the valid equivalence class, we need to look at all the characteristics of the input data just like here one characteristic was the type of the protocol used, the second is the type of the image file that is used.

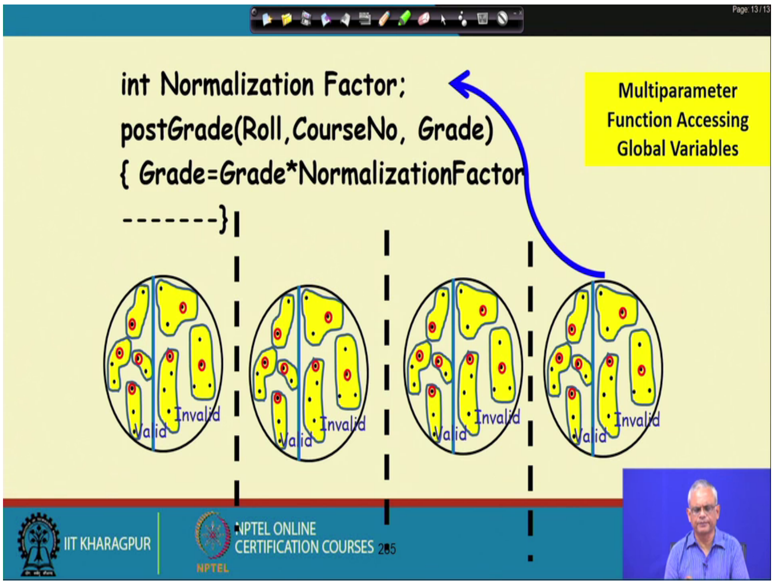
(Refer Slide Time: 12:25)



But then so, far we have been looking at the simple case of a function taking a unit taking just one parameter, but what happens if a unit takes multiple parameters? Let us say we have a function called as post grade. That is the teacher wants to post the grade, and the post grade function takes 3 parameters; one is the Roll number, the second parameter is the Course Number, and the Grade obtained in that.

In this case, we can define equivalence classes for each of the 3 parameters here for the unit, but then we need to select one of these for this one and for this one of these. But then, while giving the parameter 3 values of parameter do we test it for all possible combinations of these or do we select 1 and for a specific values of this, now let us there is a question that we need to answer.

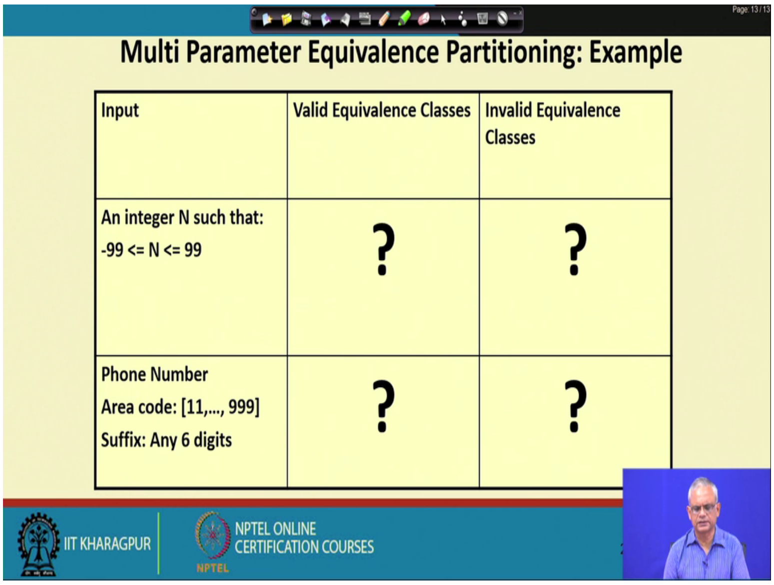
(Refer Slide Time: 13:59)



But, not only the number of parameters, but also a function might use some global variables look at the function postGrade, let us say it uses a normalization factor while computing the grade while posting the grade not only the teacher defines the roll course number and grade, but internally it does a normalization by using a normalization factor. In that case we need to consider all possible equivalence classes that may be defined for the normalization factor variable. And therefore, will have 4 sets of equivalence classes here for this function.

And, we need to consider combinations of values for various equivalence classes, but then again the question remains, that how do we define the combinations of values that we have chosen for each equivalence class. Do, we consider all possible combinations adjustably for each parameter or what do we do? Let us try to address that question ok.

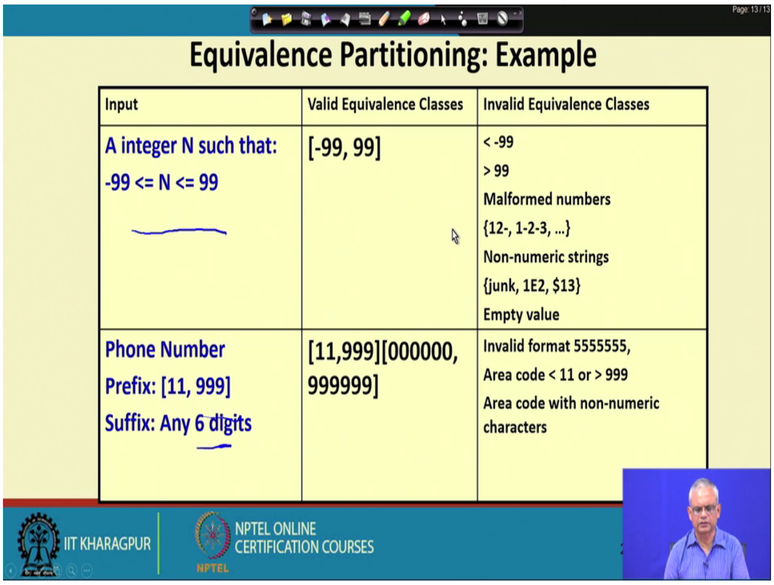
(Refer Slide Time: 15:27)



Let us see a multi parameter equivalence class example, let us say the input to a certain function is 2 parameters. One is a integer value such that the integer lies between minus 99 to plus 99. It also takes another parameter, which is a phone number; the phone number contains an area code and 6 digit number.

We know that we have to define valid equivalence class and invalid equivalence class. So, for each of these let us try to first define each of this 2 parameters, what are the valid and invalid set of equivalence classes? And, then based on that we will then try to answer that, how to consider the combinations of these two different parameters.

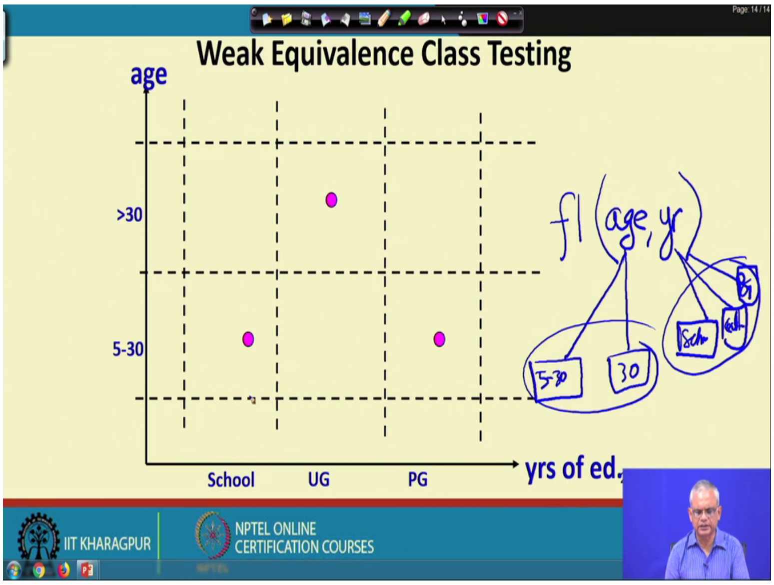
(Refer Slide Time: 16:30)



For the first parameter we have a valid equivalence class, where the number lies between minus 99 to plus 99. And then we have invalid equivalence classes like less than 99 minus 99 greater than 99 numbers which are malformed like 1 minus 2 minus 3 etcetera. Non-numeric strings empty value all these are invalid equivalence classes.

For the second parameter this phone number it consists of 2 ranges. So, we might consider that the area code is between these 11 999 and any of these it is in these correct range, but there can be invalid due to being out of range for the first argument, first part, second part. For example, we might have invalid format area code is less than 11 or 999. Area code with nonnumeric characters these are the invalid equivalence classes. Now the question remains that, how do we combine these 2 parameters?

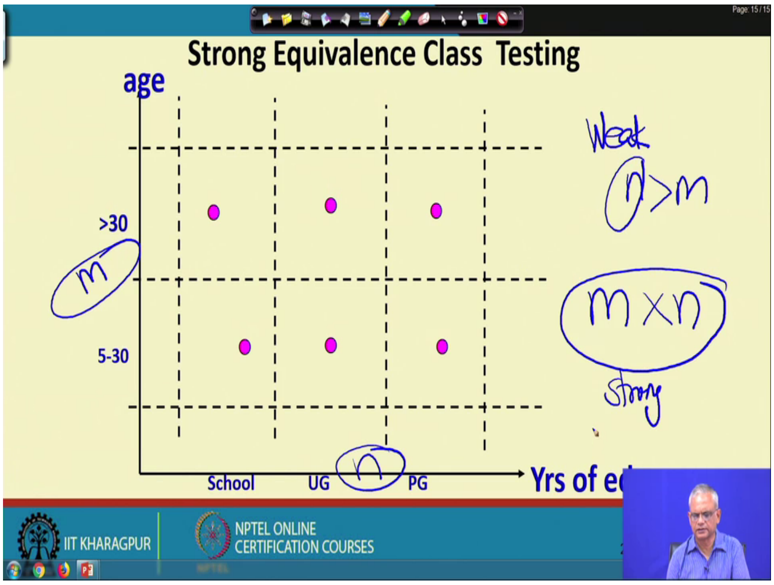
(Refer Slide Time: 18:05)



One way to combine two different parameters is called as weak equivalence class testing. Let us say that a certain function takes 2 parameters, one is a let us call that function as f 1 or something. And, it takes 2 parameters 1 is age and the other is years of education. Age is a integer and for age we have identified the equivalence classes 5 to 30 greater than 30 and for year we have identified ok, number of years of education based on that we have identified school, education, college, under graduate college or post graduate. So, the first parameter has 2 equivalence classes, the second has 3 equivalence classes.

We plot these 2 parameters here. We can cover try to cover all of these that is our test cases should be able to test both the age equivalence classes, 1 value form the age equivalence class and also 1 value of the year of education. And, in that case using just 3 test cases will be able to cover both the age valid age equivalence and the valid year of education. Similarly if there are 3 or 4 parameters, we can try to cover all of them and that we call as the weak equivalence testing.

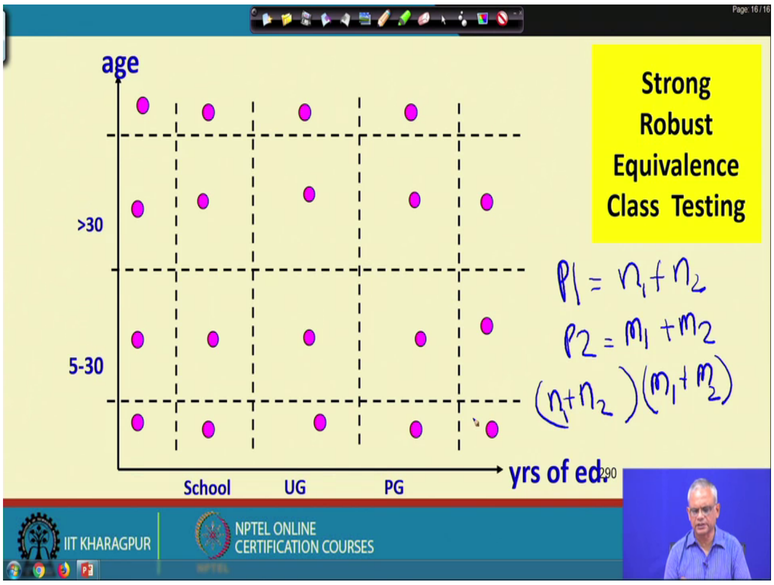
(Refer Slide Time: 20:30)



We can also define a strong equivalence class testing here; obviously, the number of test cases is much more, here not only that we want to cover all values of the 2 parameters, but also we ensure all possible combinations of the 2 parameters. So, 5-30 is combined with all the 3 of this, 3 valid of the years of education similarly greater than 30 is combined with all the 3 or if we look at another way. For the second parameter year of education, let us say school is combined with both the valid age UG is combined with both the valid age equivalence classes and so on.

So, the number of test cases required here, if the first parameter has m number of equivalence classes, and the second has n number of equivalence classes. Obviously, we need m into n number of test cases to achieve strong equivalence class testing, but what about weak equivalence class testing, if a 2 parameters with m and n and let us say n is greater than m, what is the number of test cases that we need? We may just think about it, but then that is equal to n. So, this is the number of test cases required for weak equivalence class testing. And this is the number of test cases required for strong equivalence class testing.

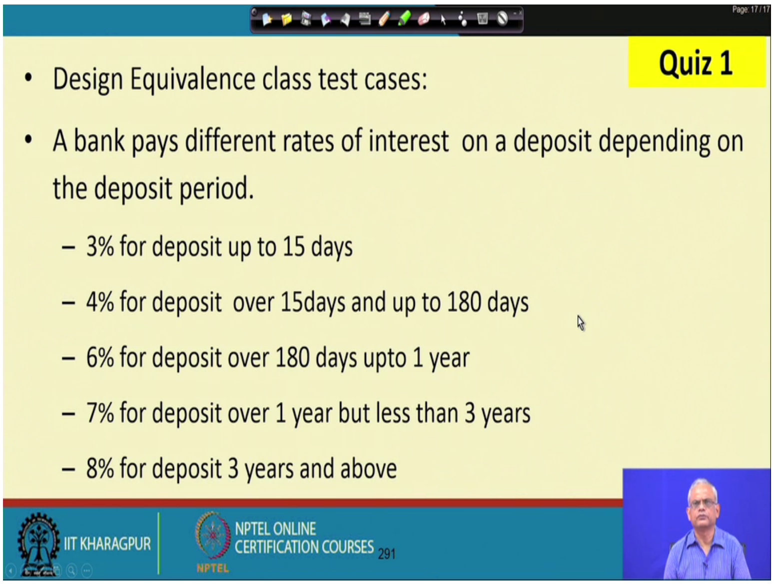
(Refer Slide Time: 22:37)



But, we might also like to consider the invalid values for both the parameters. And, then we call it as robust equivalence class testing. And, here the number of test cases required is total number of valid equivalence classes for 1 parameter plus the number of invalid equivalence classes. And, similarly the number of valid and invalid equivalence class for the other parameter. If for the parameter 1, let us say the parameter 1 there are n 1 valid equivalence classes and n 2 invalid equivalence classes.

And for parameter 2, we have m 1 valid equivalence classes, and m 2 valid equivalence classes, m 1 invalid and m 2 valid equivalence classes. Then for robust equivalence class testing we need n 1 plus n 2 into m 1 plus m 2 number of test cases and; obviously, the robust equivalence class testing will expose many more bugs, then a weak testing or just strong testing.

(Refer Slide Time: 24:20)



Now, based on what we discussed let us try to solve one example? Let us say we have a problem, we have a function that computes interest for a certain principle. And there are different rates of interest on a deposit depending on the deposit period. So, we have this function compute interest, it takes let us say deposit amount and a period.

And we know that the interest computed depends on the period, it should be 3 percent for deposit up to 15 days, 4 percent for 15 days to 180 days, 6 percent 182 180 days to 1 year, 7 percent for deposit over 1 year, but less than 3 years and 8 percent for deposit above 3 years. Now, observe here that even though there are 2 parameters the behavior of the function does not matter much and the amount. So, amount there is only 1 equivalence class, 1 valid equivalence class and then we have invalid equivalence class.

Whereas, the period there are 1 2 3 4 5 so, there are 5 equivalence classes for the period, because depending on the period, it provides it different behavior that is different interest rate. Now, the number of test cases required here is for let us say strong equivalence testing is equal to 1 2 3 4 5. So, that is the second one. For weak equivalence class testing, it is also similar because the first one is one.

So, both weak and strong equivalence class testing will require 5 test cases, but for robust testing we have 1 valid and 1 invalid for this so, there are 2 classes for the first parameter. The second parameter has let us say 1 invalid class. So, we have 6 here. So, the number of test cases becomes 12 for robust testing, we are almost at the end of time for this lecture.

We will stop here and we will see that there are many more strategies for black box testing we have just examined 2 simple strategies. One is the scenario based testing, which is a the possibly one of the simplest black box testing strategies and we also looked at the equivalence class partition. We will look at few more strategies for black box testing in the next lecture.

Thank you.