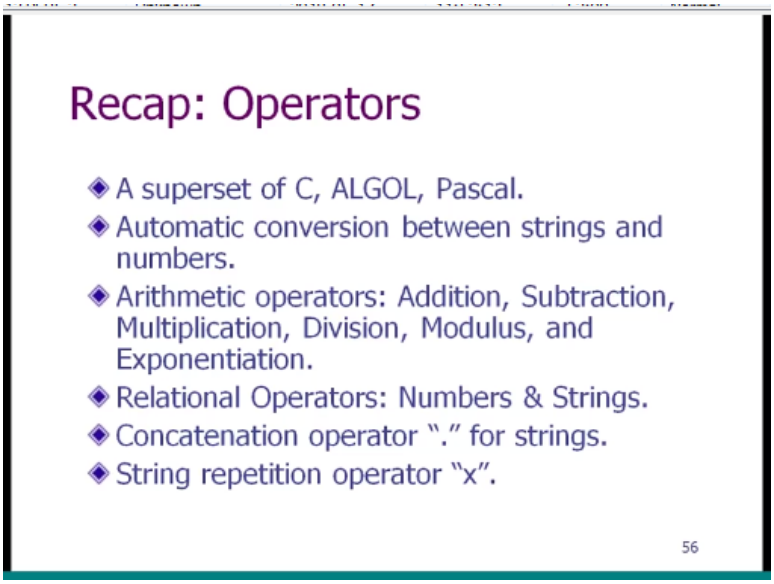
SEER AKADEMY
LINUX PROGRAMMING
AND SCRIPTING –
PERL 4

Okay again welcome students to do another lecture of Linux programming and scripting, today we will be continuing the Perl last time we started talking about some of the concepts of Perl just slightly advanced, then what we talked about before so we will be continuing the same discussion today, what I will do is I will start with the recap and then I think you remember I assigned it with a simple problem to you which I wanted you to experiment in point find out the result problem.

So I will explain that problem and then we will continue on with todays topics, today we will be focusing main form further some of the strings and some scalar operations and more from some more commands that we using again.

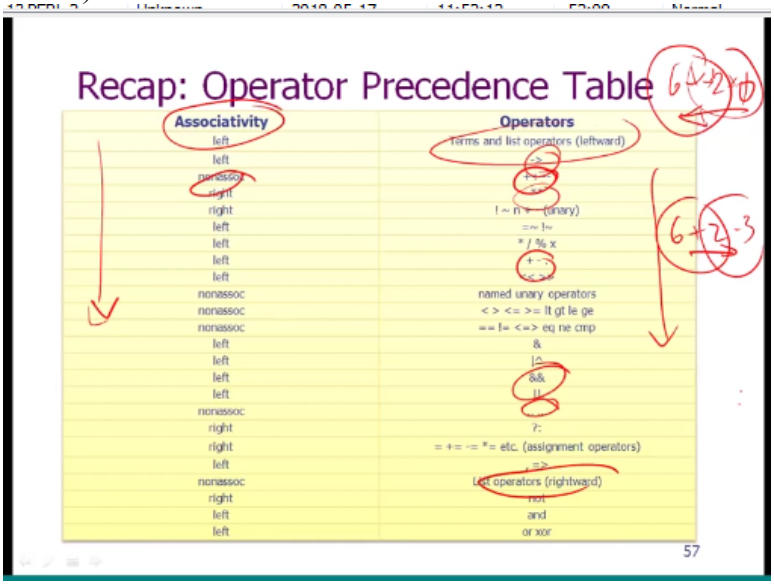
(Refer Slide Time: 01:16)



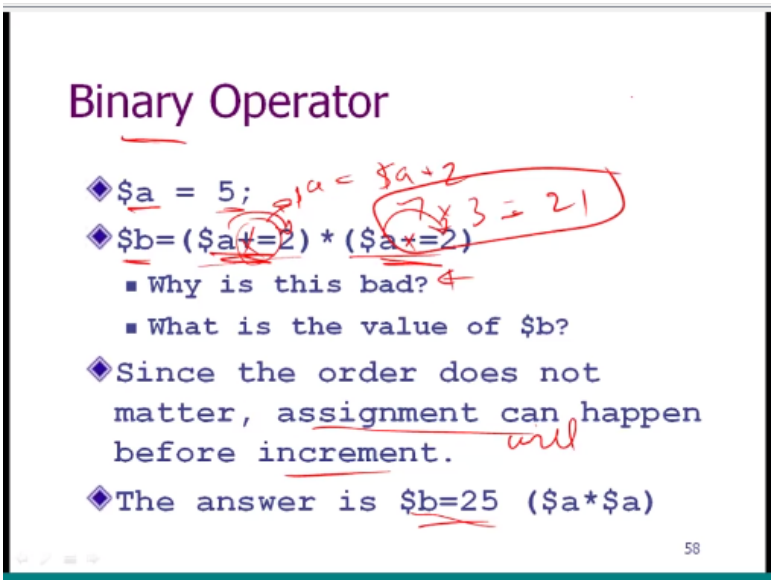
Let us look at what we did in the last lecture we studied about the operators actually and then in Perl the operators are superset of C Algol and Pascal operators, the reason is an automatic conversion between strengthen mumble or the strings basically the number part of the string is actually kept, and the remaining portion is just turn to null and then the operator is wise actually there are many operators or main ones are addition, subtraction, multiplication, division , modulus and mechanism these are all the medical taters.

Then we have a bunch of relation love is that bow between the mumbles and more rings these are less than greater than =, not = less than or = greater than or = things like that we already saw that, and in table as to how to use the relation operations operators and then, we also like basically

what is the difference between we have a number as a variable our string is available then there is also this concatenation operation operator, which is prevent is dot so that is that is a another one yeah.
And then we also saw this in repetition operator with the X essentially, this one actually when you say they say like in the string to name and then more below name times three is like the naming name to and then we saw like I mean how that works being think one of the key concepts that I introduced the last lecture was this concept of the operator presidents table.
(Refer Slide Time: 03:19)



Here the presidents first one between the various operators it goes from top to bottom this is the highest presidents essentially, operators and then basically your president so you these operators come together first the fire president evaluated before evaluating any of the moves, that is the first rule that we saw and then between the operators it goes by the Associated, so for example like an think above exponentiation operation and you have multiple ones like a 6 6 exponential and 0 2 exponential 1 0 so how the way that it evaluates is from right to left.
So we should be when this works and then it is an expose whereas an operation like the + or minus or multiplication, they will be like I mean 2x + 2 we go together in first and then, then so it goes this way, so that is the other, other, other key aspect that you learn, this way we are able to solve, so some of the things that are that why I want you to pay attention to our one is non-associative.
Once this is the increment decrement which is what we will be about today also like these kind of things list operators so those are like the non Associativity video so pay attention to these essentially like an important as we go, anyway so now we will be looking to the next form.
(Refer Slide Time: 05:26)



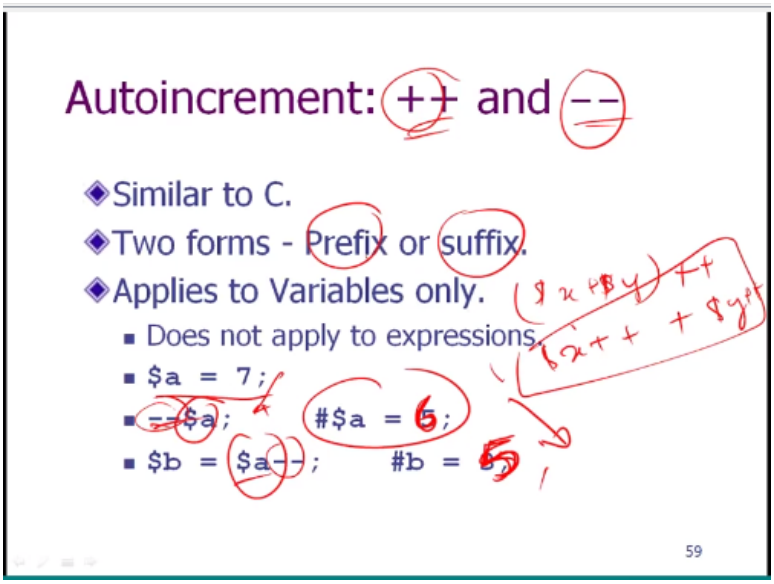
So he actually liked I mean I asked you one question where you find that $ A F5 and then B is actually more using this binary operator which is 2 operator combined + = its traditionally represents a $ a equals $ a + 2 but now if you do this why is this bad and what is the value of Ecology this is the question that I asked during the last lecture I want to give you the answer today, the number one is essentially why it is that is because what you expect will never be the result of exhibition.

So here you may expect that basically this will be is actually seven times three should be 21 but that is not the case so why is it not the case and what is the actual value so before that I mean if you look at the presidents rule and say it is basically in this kind of scenario the order does not matter, so since the order does not matter assignment can and in fact will happen before the increment, so what that means is A is assigned A value of 2 and here also A assigned 2 whereas the increment operation system it will.

It happens but those results are not saved and the actual answer is this 25 which is quality time delay or people who have already experimented with this, and got this answer that is great for people who have not gotten the answer I will try to ask you to try to experiment this in your laps, and see whether you are getting this answer, if you do not get this answer please let me know send me an email with what the answer that you want, and then we can discuss what is the but anyway the bottom line is doing these kind of inside tricky and never use these kind of syntax.

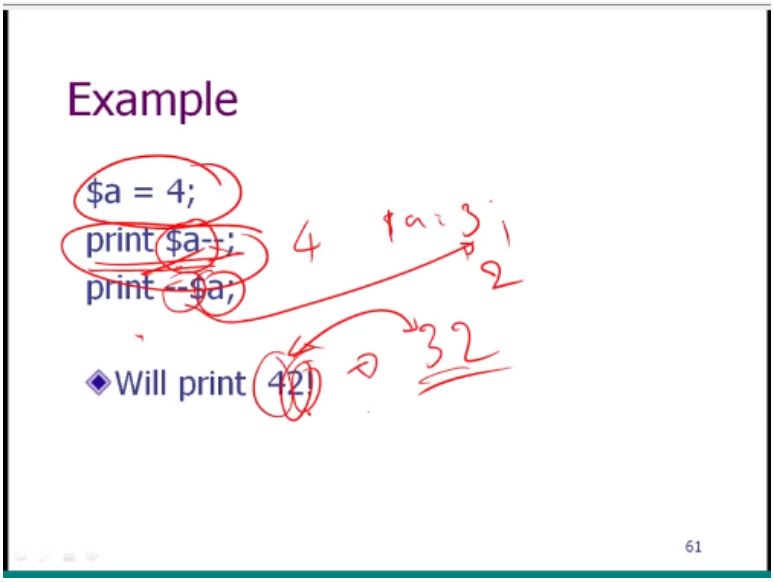
Unless you are ultimately you are absolutely sure that, that is what are intended to so, that pretty much covers what we did Dustin, so in today and will actually talk about more sort of binary operator we have talked about unary operator which is essentially like auto-increment and not

too dependent, and then I will talk about some of the caveat on that and then we will go into more interesting topics, and I am talk about some of the things as to how you can manipulate the programs for problems so without what you do let us look at this lecture.
(Refer Slide Time: 08:41)



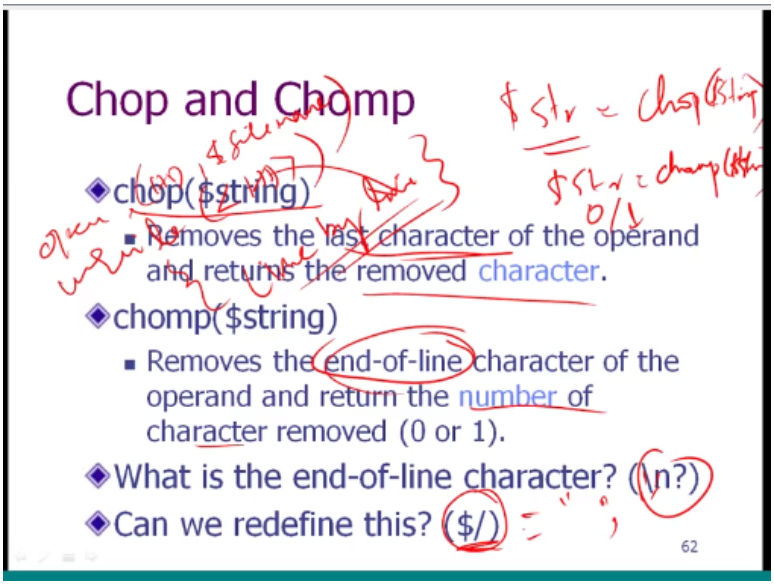
So the first one is the auto increment call it auto increment so like both + + this is the auto increment, but these are all the unary operators meaning that they operate on one variable you will be the second variable, and these kind of operations are very similar oh you already followed I mean see there are several of things like your I + + things like that that we always give within the loops and things like that.
So they are very similar to see again 2 forms are supported one is a prefix and the other one is a suffix the prefix is when you have the operator in the front and then followed by the variable or the suffix is essentially like a mission or the operator is after available, so here if you look at it basically and then one, one thing to notice these operators are applied only to the variables you cannot apply this to an expression make all of X let us $ Y and then you cannot say like a moving loss, loss something like that.
So you need to make sure that reduce this thing and then do all you can do make them well the X 10 X + + and then + oh, why this is a valid syntax but when you write this kind of syntax there are some issues so we will learn about that, but before that let us look at this simple example is assigned at 17 - - $ 8a you actually should sit so it implements by 1 and this should be 5 sorry for the typo so and then B is 5.

So here essentially like a minute is decremented by 1 and then again none of the decrement 1 so this is the result, but now if you use this and actually there is a difference between the prefix and a suffix, so let us see like what is the difference prefix and suffix.
So here are some of the rules basically like a used within the an expression, there is a difference between these two expressions these two values so if you are just using it as a standalone like this, it is okay basically they still retain two value will be decremented the value in A so the prefix and here it is basically like is finally end of the operation is actually like this the decrement and for the decrement values assigned to P so it is still 5, but if you are using inside an expression then there is a difference between like 1 this first case essentially that is listed here which is the, the value of A before it is incremented will be used in the expression while for the second one the value of A after the implementation is actually used in the expression so let us look at one small example.
(Refer Slide Time: 12:13)



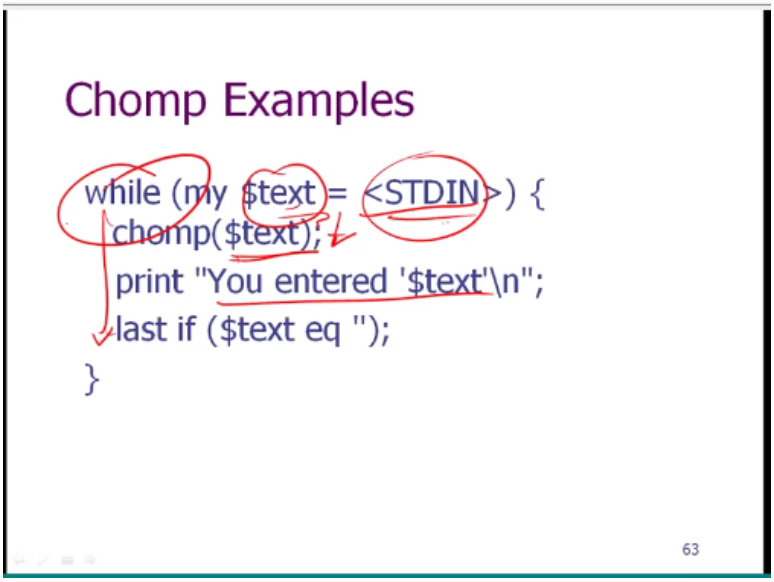
Here A is assigned as 4, and then we basically ask it to print $ a - - and then print - - $ A so what we get is actually like 4 2 or for the expect that this should be like 3 2, so the reason why this form is unchanged is because it basically like this treats it as an expression and it prints this value before actually like doing the equipment, so basically like it prints the value of 4 and then it decrements the A so now the $ A is actually 3 at the end of this expression.
But now when we go into the print next one, the, the 2 negative sign actually decrements before the 2 and then basically like that the value of A retains and then that is what it gets printed out which is 2, so it is instead of 3 2 actually get O2, so again understand the consequence

understand how world behaves and then you will be able to actually exactly do what you are intended okay.
(Refer Slide Time: 13:36)



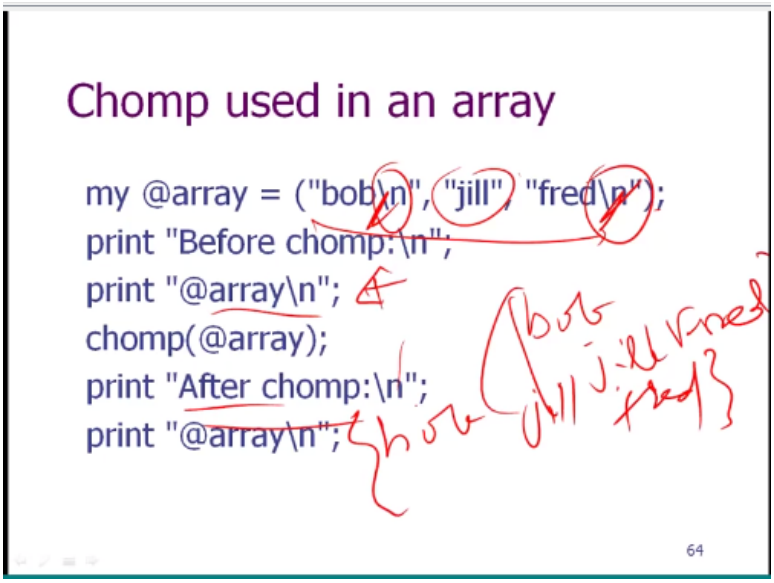
We will again revisit this concept program but I want to leave you maybe leave you with this essential, so now there are two important functions, that we will learn I just wanted to highlight the difference between these two functions, body that actually do they do first of all then what are the difference between those two functions, and then how to use them in various code segments, so I will illustrate through the code segments the 2 functions that are kind of integral to purl are this chop and chomp.
So essentially like I mean they are listed here the chop essentially removes the last character of the operand or of this argument and then returns the removed character string, whereas a chomp actually remove the end of line character this is very important, it has this end of line character and then it remains it returns the number of characters removed, so if you do like I mean say like $ STR = chop $ string the STR will have the string with the last one removed, whereas if you do the $ STR = chomp $ string this one you will get this either STR will be a 0 or 1 based on whether chomp remove the character or not.
So that is one of the key differences and jelly so one question I have is what is this in the comparable, this fixed one make them in for example back/ in that we used in previous classes, how do we even change it, so can we redefine the inter plane tag because one of the things that is kind of we will go into is to read the file, mainly the Perl programs have a very similar syntax,

basically they will open a file using a file handle so open any file handle this and then what a file name.
Once you open the file name then start reading the lines in cycle 5 which is using a while y comma HB so it reads line by line, so now we have this line by line that is what is a little different, but can we change this to some blocks, acting so that we can input the whole block inside how do we do them, so yeah in Perl actually provides a lot of special characters one of them is $ / the $ / is actually that stores the end of line have , so by default $ / = current oh sorry back/.
So now you can see actually this back flashing you can change which other character that you want say like I mean you have the semicolon delimited file, you simply change this so then it, it treat semicolon at the end of nine characters and then it reads all the way up to that semi colon, so processing like blocks of data like say like a test file or something these commands are much more useful you understand that the $ /.
If the character that denotes the end of line then you can easily go there and modify that and say that okay end of line character is actually this period that means that it will read every sentence whether it, it does not matter whether it conforms to just one line or multiple lines it reads every sentence, so again easy way to actually alter the behavior.
(Refer Slide Time: 18:48)



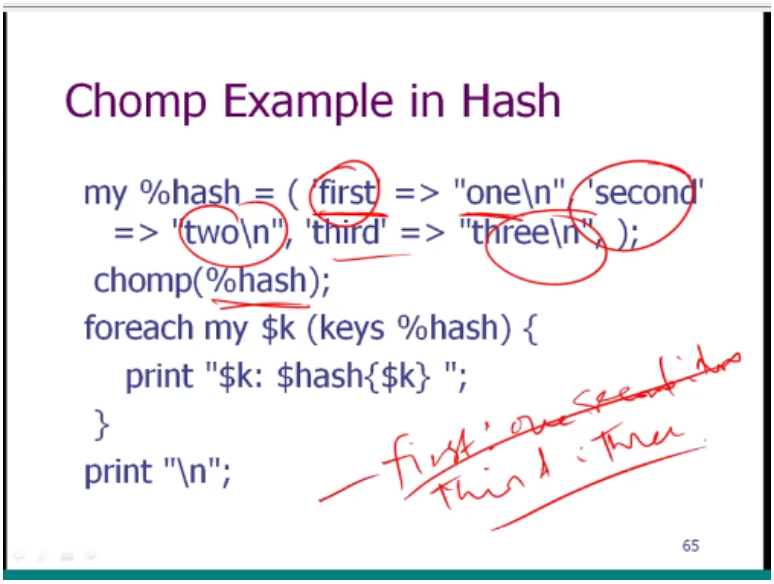
So now let us see like some of the examples of chomp again I, I did not want to actually put examples for the Chop function because the reason is even though the Chop is also used fairly closely like been similar to the Chomp, Chomp is more interesting chop is essentially looking

just remove the last character, and it gives you back the string so, so let us see like the some of the examples of charm.
So here is basically it is making simple more text read as I mentioned the same thing basically like them in this case like in here it is going to eat from the standard in this we will see in like later on how we can do it, and then essentially the sports segment and say it is essentially our own variable that we assigned to stand it in so that waits for the standing input until we press the new length that STDIN goes into the next moment.
So here the chomp $ text is essentially like a minute is taking this form and then just removing the end of line character from the end, as you know like I mean feed into the standard in we have to press the enter in order to go to the next step, and then in this one prints out the, the $ and this continues on until we put nothing and then this is painful so this is a one way of actually looping this is most common loop is the while loop pearl and for anything and everything will most likely use a why you because you do not know then the file will in so we operate on line it is time until the file is so we still use that while you in why is one of me work causes of Perl problem.
(Refer Slide Time: 20:48)



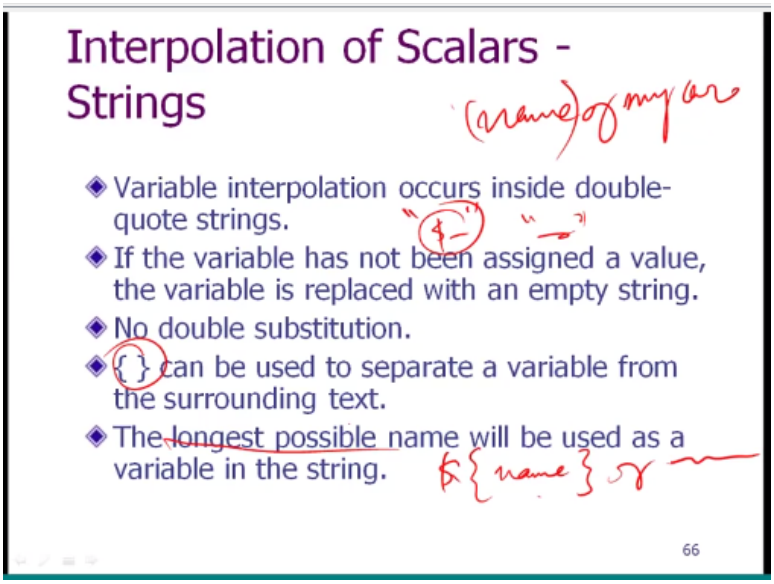
So now let us look at another example then this chart is used in an array AC so in this array you can see that actually bob has a new line character, and Fred has a unit capital so when we do like a print before chomp and then array it distance exactly the only thing is still is coming in the next nine and game I spread and then in it one line pointed, so it will be like bob Jill and Fred , this forms one ratification so this is usually yeah okay chomp array sorry okay.

So this is the same as here and so now let us see like I mean what happened with chomp on this array when you do the chomp on the array actually get the to do line for economic and the array will simply be Bob Jill and Fred, so and this is probably what we want so that is basically like using the Chomp, Chomp.
(Refer Slide Time: 22:15)



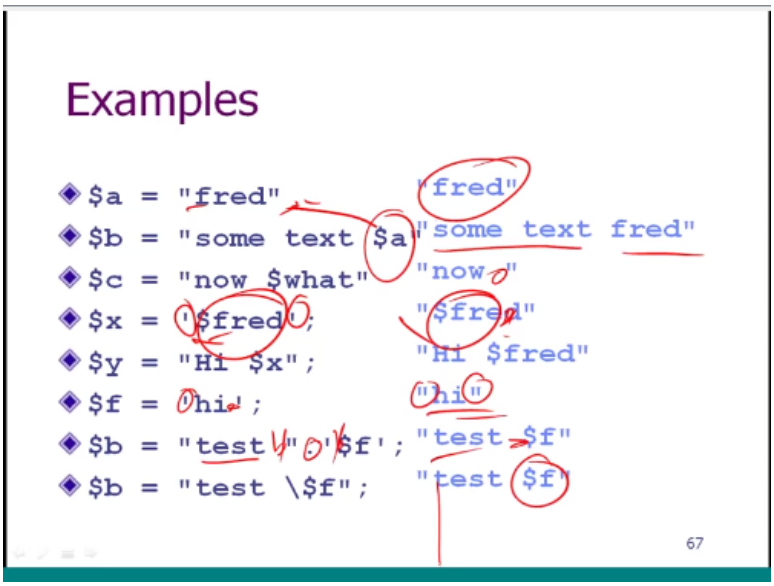
So now there is a third use model which is actually using the, the charm in the hash again in this case actually like hash is something that we have not talked about Hash is very also like similar to concept to the C language a hash essentially a labeled or a linked list of not even linked list this list basically which has an ordered pair concept which is basically it is similar to an associative arrangement A in fact it is an associative array.
So here, for example in a sorted array we saw like I mean I really want to specify even number of elements and then the asset post again just with the second, second the third I am sorry third to the fourth and fifth to sixth kind of thing so here you can see like basically first is actually as with one at the its main value the hash itself is the first, and then the second has two at value and then the third has thirteen I mean three as a value, and everything is followed by the new mental.
And then once we do the chomp of the hash then we can directly get this one this, so it will tell you like I mean first all in one and then the second sorry second all in two and then third and three this these things will be in the same on, We have in modify and then after that there will be printer you like that so I hope this one is clear essentially like I mean so the chomp essentially like I mean eat versatile function you can use it in many, many, many rays oh.
And named inside the, the core program the three main base of using chomp are mentioned here, and then essentially basically what it does is it removes the evil characters from within the thing

and it gives you the results after removing what it returns itself this is for meaning is we all want based on how many character removed but at least, you will the updated string is kept in the original variable itself about that I hope this section is clear.
(Refer Slide Time: 25:19)

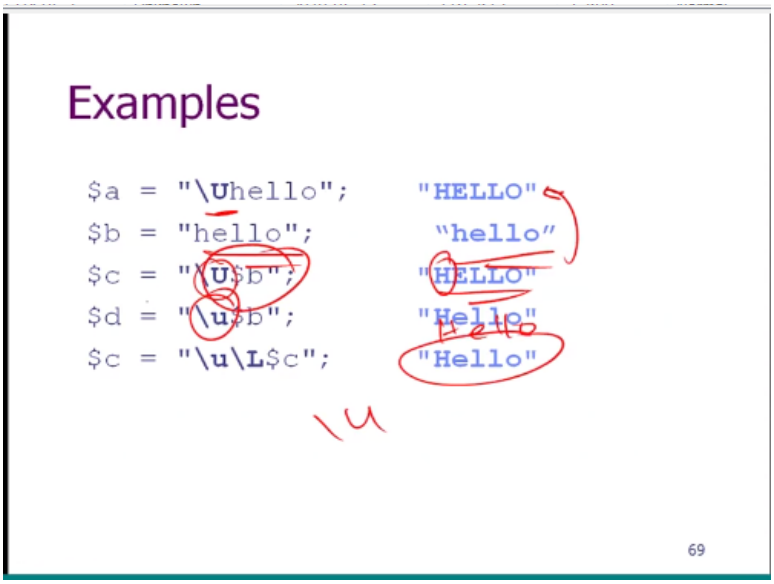


Now let us talk about the interpolation of scalars and strings so the so you know within a string like a manicure using more variable with a scalar or string that is evaluated inside a double quoted string, so if you use the double quoted that you already know then you know that basically they the double quoted strings inside any like other variable is all value, if the variable has not been assigned a value variable is replaced with an S an intestate, so if there is no value to this just basically within constrain nothing but.
Now there is no double substitution within the interpolation so if you have like the nested in basic in $ XY then now actually like further down all those things basically those nested ones are not allowed, so you cannot have like another quote and then something else and then in code and use it this exercises that 10x more below what so, so that is another room and then finally, I mean the longest possible name will be used actually sorry I missed one thing.
So we can use the parenthesis to separate a variable from the surrounding text so for example you have like the name of my or this is the long variable, or you want to just say like I want the name of the variable, so if you want to just as are asked for the name, then you can just say like $ in parenthesis name and then off my top so this is one way of doing it and we will real actually like go through some of the examples that they make this concept very clear and then the other

thing to note is bottom of the longest possible name will be used as the variable inside the string, so if it is in our table it just goes on and on until fine.
(Refer Slide Time: 28:01)



So now let us look at some of the example of strings and the scalar variables so, so here there are some examples so $ A is assigned to Fred, and note the quotations there and then we say basically like some text A for what will happen, so here is a sign read so we say like these some text and by the way that some text Fred, so this is even greater now you say like now what since the $ watt is not defined that is left as an empty string so it's basically just now.
Now what we are doing is you are using a simple code to assign freed to caller X, so one thing to notice in your using the thing you put only the back/ and single quote itself can be escaped all the others you do not need in this case, so you can unity with keep A $ it is going to print $ set so let us look at all so you spinning bubble now we say like okay $ Y is high $ X in this case what it prints out is the whole $ Fred because $ Fred as the value of that particularly.
And then here again a single coat with the space on the space is much recent meaning you get double coat and high, and here it is a concatenation operation essentially so basically like push test followed by applying and then dot and then be like followed by another yeah, so there are two blank within before test and then the last but not the least test escaping the $, basically it prints out a smaller F.
(Refer Slide Time: 30:34)



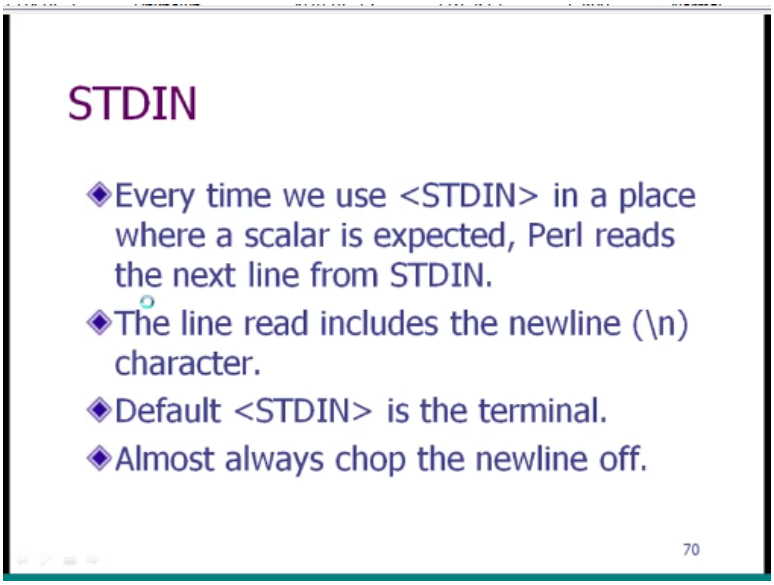
So let us look at some more examples this is kind of interesting one softly oh yeah same $ sun to A and $ Sunday is = wrong, so then we say like I mean $ X = it is a Sunday it is just say like it is wrong, now the next one is Sun space day then if you say it is a pay day the Sun is a sign to pay and it is this is the if you put that inside the parentheses this will have the same effect, the only thing is like I mean now the blank spaces are all gone so right next to it so for basically the basement would not sleep at all.

So it just spins out at it is Sunday and so the here the this, this variable is being replaced and then now the concatenation operation essentially no home for the $ son because within the double quotes that is expanded and stay, and then since there is a concatenation operation this day, so we still get it fading this one so that is one thing, and then finally like I mean we can also split this in multiple lines and more this operator can be used in many male multiple times concatenation operation.

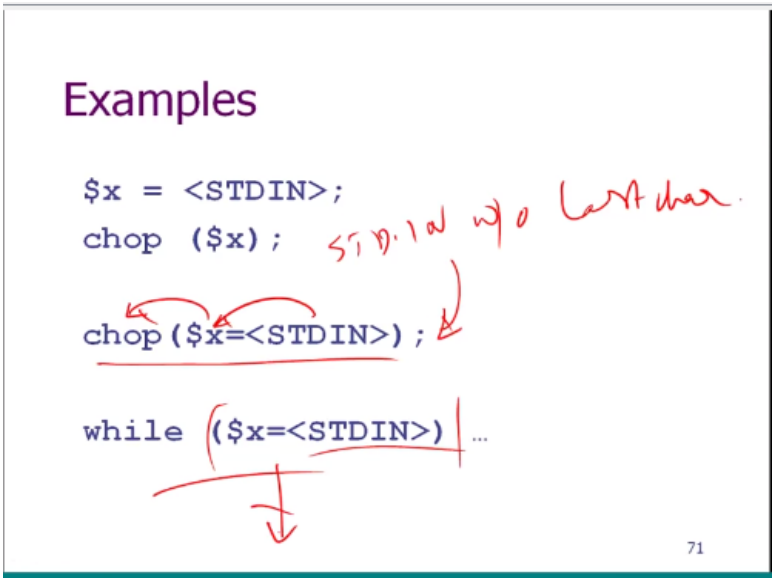
So that means that it is a if and then quote which is concatenated with $ sign is contended with just the day so finally you will get this result it is a fading its fading, so now let us get some more examples essentially you know these are other escaped characters that we kind of touched upon it but not really long one is the escaped uppercase you this stands for convert that everything in this particular string interval case oh, that is what happens.

And then we assign me to just hello faith along with the same here and then if we apply that same transformation like the back/ you on the entire variable B then we get the same a it is hello, and then if you use a lowercase with you then it only changes in first one and then it continues on with this one h-e-l-l-o let us see and then finally like I mean if you do a replacement of you

uppercase L and then with the same as this other hit hello, hello here basically like we find that actually get the same hello you see this okay.
(Refer Slide Time: 34:27)

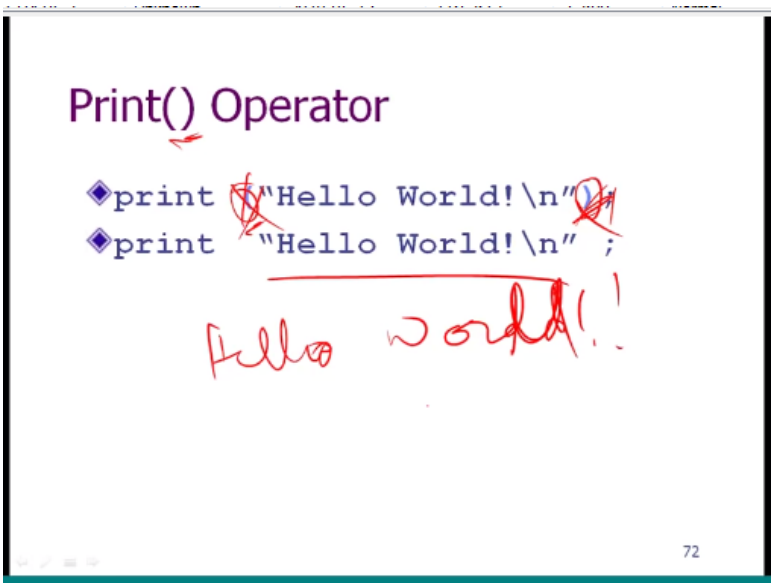


Then now we come to another important item in the pearl which is the standard in as I mentioned essentially standard it for getting the values or getting the input from the terminal itself so the key thing is like every time you, you can stand again in place of a scalar in place where the scalar is expected the pearls read the next line from STDIN so this is one way of doing an interactive session there you can use a STDIN to import value from outside words.
So the typically the STDIN reads as an eye and line and so it reads the everything until the new line is it is expressive, so and it includes that new line character so you need to do like a chop or a chomp and to remove that new line like that, and then usually the default STDIN is the terminal that you are used to in so that is called the standard IN.
(Refer Slide Time: 35:52)



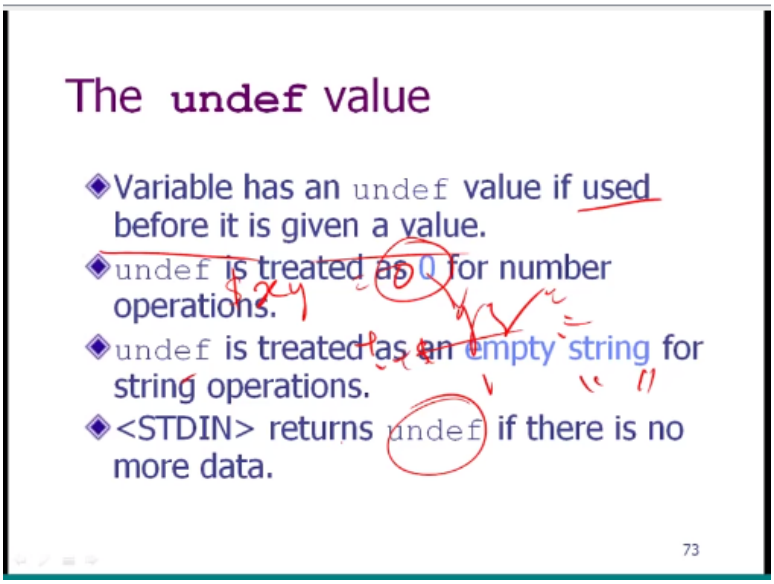
Let us see like I mean so some of the key examples here, one is we assign the $ X or STD in standard a into $ X and then we do like a chop $ X and then here what we get is essentially the STDIN without the last character so they like STDIN, another shorthand form or writing this is for you directly through the chop, so in this one basically like you have to look into the presence metrics, but first assigns this and then chops resulting value, and many people just write like this standard crap fact it because this also gets the line by line and then for every line immediately like a passing this loop, and then pending suit else it executes the loop.

(Refer Slide Time: 37:15)



So the next operator is the print operator, this is also like something that they are familiar with to induce all the familiar topics first before going into much more complex and advanced topics, so

here a print operator observe that they have a parentage and we actually put this parenthesis, but it is kind of redundant basically like for like a standard package people going for this, this will be directly like whatever they want to write, so here basically like having this particular program distance of hello world.
Whether we use the parentheses or not the outcome is still going to be this hello world, so pretty much like I mean Print is fairly easy there is one more thing with you for the format we will talk about this in one of the other lectures, at this point I want you to understand just the basic print command, and how to use it within the, the file itself this is so, so on one hand we use the 18 iron for taking the input from the terminal and on the other hand we also use for print to print to the terminal so, so those two commands go hand-in-hand now the next one next up it gives this undefined concept essentially.
(Refer Slide Time: 39:05)



So essentially like I mean the if it is a numerical variable disorder actually go between basically an empty whereas in a numerical value the variable has value status under if you use it before it is given a value, so what is what is mean by use this for it is given evolving, so every variable has some value that many beneficent so as a standard practice we can say like an one of X Y 3 = 0 um if you do not do this and then say like I mean using $ a $ ABC + $ it is like Z = LMN.
So this element = $ so this is maybe a go this is one of the things and then the XYZ and then at this point like if $ XYZ is not defined and only the $ ABC is defined, then you get an undervalued or this in this case actually like you have defined this as 0 maybe next okay but ABC not defined, so this becomes on this and unless is typically treated as 0 for all the number

of operations, so even though it is and if the resulting answer like them input will be distance 0 times of file.
So this is whatever the values you, then for string operation is threatening to string which is all the coding I will call strings so it is used as that that one now the STDIN actually returns this under there are normal data so essentially it goes into the loop it reads all the file puts on them listen, if it reads that essentially whatever providing it from the STDIN and as we hit enter it goes on, until we hit like nothing basically then that returns and on this value okay.

So I think like I am going to stop at this point and this to do a recap of what we did today, and then we can actually come back next week the next class, and then you can pick it up from this point, so today we focus mainly your more repair you remember all the things there are two main topics essentially, one is we talked about this Auto increment and the Auto increment and how do we use them in the overall programs, so this is a handy way of doing it most people use it inside the for-loop okay.
So for looks, looks like when I get $4 I to : is 1 10 semicolon $ I + + and then, so this $ I + + is one way to use it on left and then you know the difference between goes, so then so those are part of this rule, then we also went through the chop and the chomp functions chop mainly work with strings and basically in blue or any strain, whereas the chomp packs we have removed the newline tag from within the page so and then the newline character itself can be modified.
So they mentioned the unity the newline character back/ N is actually set to this variable $ / so we can capture this girl / and move to the third one that you want basically and then we can now use that for as the in the interfile, and then we saw the examples of chomp across many, many different disciplines for example, here it is basically just a standard way of using the chomp just applied to particular string variable.
Here the same chump is used inside an array how we can make it better, and then the finally like I mean the how do we use chomp inside a hash function and then I begin this, this hash to you essentially it is an ordered pair basically and all these things as a hash and then first equals and this can go like I mean maybe think , we should be making will not be this passion of the first but then you can think of a for using this that is and we will visit some of the main principle.
And then we also talked about the interpolation of scalars and strings essentially where we said that, basically the variable interpolation occurs inside the double quoted string so that those are the main, main things and then if the variable is not be more find a value then it is replaced with an empty string, and there are no double session things in meaning that you cannot want to

transform many times or different transform is different times within the same string, and then this pandas will be used to denote that which one is the actual variable.

So here you can say like name of my car something like that and you are only interested in the column name, then you can just say basically this stance is inside and off whatever and that essentially, yah gives only like the name as other nothing cancels, and then one thing is also like I mean once you do not have this then it uses the longest possible name to be used as a baby exactly this one if you would not use this fantasy then we go like all the way down from this.
And then we saw some of the examples of for this string manipulation, you remember these things, and then we also saw some substitution characters like the back / you this stands for uppercase throughout so it takes the hello in lowercase, and then push it out to so this is something like the transliteration example that we saw, a couple of classes ago, and then here again the same thing skipping the column or not, and then here essentially it will basically like only change the first character of the $ B the remaining are left in time.
And here essentially like a change the lowercase numbers to LSC of Illusion actually code, it is the it replaces the U with L, so basically you know that is, that is the effect of this one, so we will see what you so it basically is the same value on this one, because originally it was all like uppercase or basically like it is the first character has the same page a page and then it went in lower case on which is losing you for a uppercase cell if you use a lowercase L then it will only like lower people person to first character and then the remaining ones you can leave them as is.
So I want you to understand that also, so and then we also saw about the STDIN how we use their STDIN and the basically like different days of for combining the STDIN with chop, and then this will be how that the hello world using the print operator now this is another one, so in the print operator right now being studied basically just how to print a string, there are also like formatting options which are available, and then we also can assign a return code help the furnace, so these two topics we will be talking about it as we go along.
Because these are important for other, other purposes also and then we also saw about this, unless basically what happens in particular variable is not defined ,out its full react and basically most confidential undefined value, so I think that is pretty much it for todays lecture we will pick it up from next week I will be assigning some more segments essentially so that you can practice your poll, so far what we have learnt in terms of how to build a better program so using Perl language okay thank you very much.