



IIT Madras

ONLINE DEGREE

Computational Thinking
Professor Madhavan Mukund
Department of Computer Science
Chennai Mathematical Institute
Professor G Venkatesh
Indian Institute of Technology, Madras
Subtypes of basic datatypes

(Refer Slide Time: 0:16)



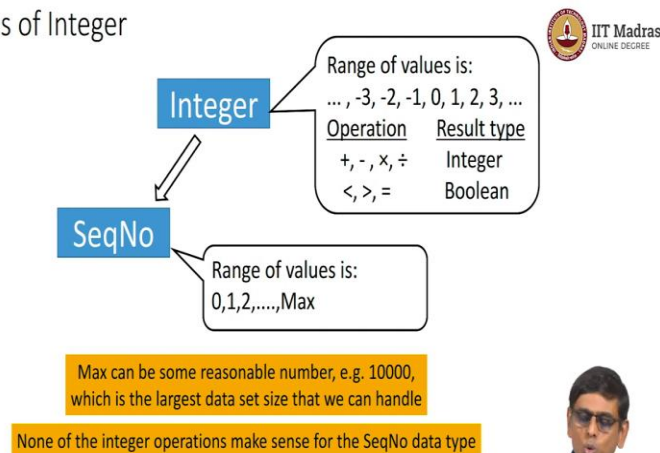
Subtypes



So, the next thing that we want to look at basically is once we have defined the basic data types of Boolean which is true and false, integer which has values which are negative numbers, 0, or positive numbers and character data type which can be alphanumeric or special characters. We want to see whether we can restrict these data types to get further narrow ranges of values and further constraints on the operations so that what we wanted for our data sets is implemented.

(Refer Slide Time: 0:47)

Subtypes of Integer



So, let us take an example, the first example that we can take is the sequence number that we saw on our cards. If you remember all of our cards had a sequence number on it and the sequence number was a valid number which was starting with 0 and could be any positive number 1, 2, 3, 4 and so on. But clearly a very very large like 1 million of something is not a meaningful sequence number, very unlikely that it will be such a large number.

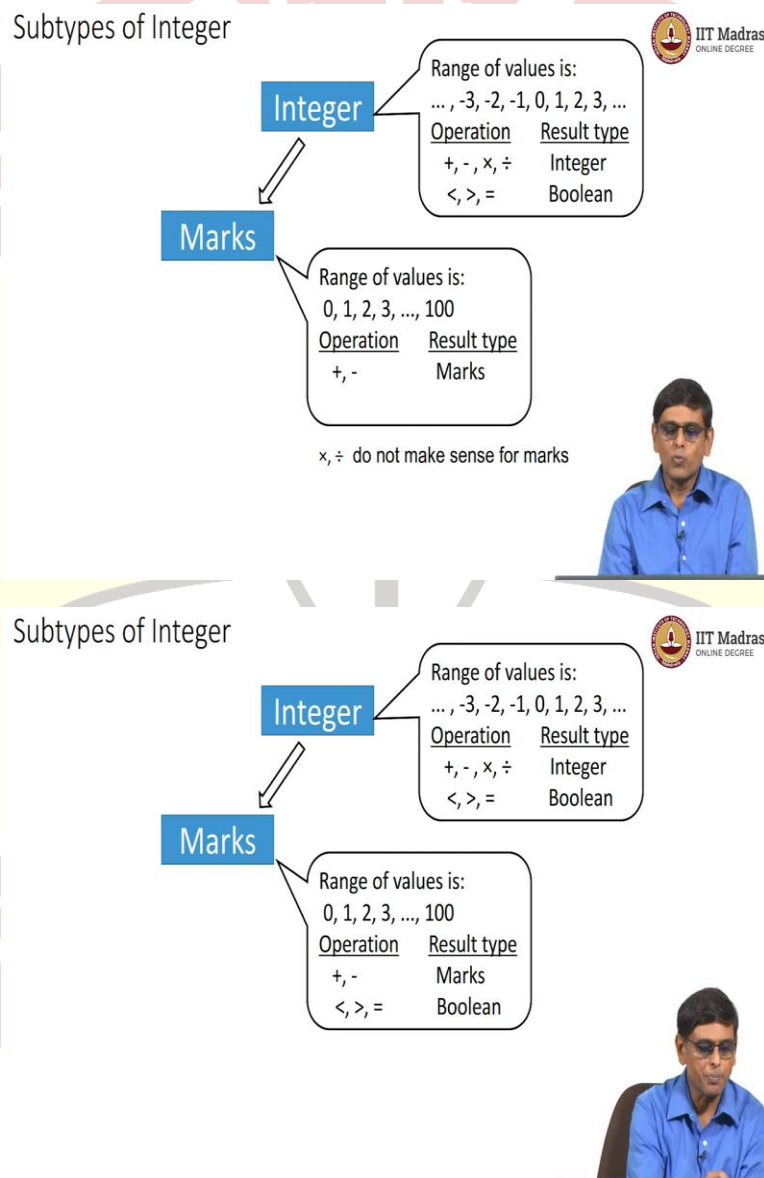
So, we could take that the valid range of values for the sequence number is 0, 1, 2 or any number like that up to a maximum number which is max, and max can be some reasonable number like say 10000 it can be for example, which is the largest data set size that we intend to handle. So, in our data sets that we saw basically we had 30 marks cards, we had about 50 or 60 shopping bills and so on, so we had even less than 100, so in our example set we could even set the max 100.

But you know computers can handle much larger numbers than we do physically by hand, we would like to keep the data set small but when computers can basically handle even 10000 or hundred thousand or more. So, we can keep max as some large enough number, something like 10000 might be reasonably ok, in which case basically the range of values that sequence number is expected to have is anywhere from 0 to max.

And in terms of operations none of the operations of integers seems to make sense. Like addition, multiplication, those kinds of operations do not seem to make sense for the sequence number data type. Maybe we can do comparison, you want to check whether one card is coming before another

card and therefore you could use the less than operation and you might use the equal to operation, but at least the addition, multiplication, subtraction and division operations do not make any sense for sequence number. So, you can take the range of values as 0 to max and considering the operations to only let us say the Boolean operations.

(Refer Slide Time: 2:45)



Another data type that we saw as a subtype of integer that we have been seeing is marks, that we saw on the cards. So, what is the range of values for marks, clearly be in our data set at least we saw that the marks basically was positive number, so the negative numbers are not allowed. 0 is okay because somebody can get actually 0 in some subject, so 0 is okay.

But in terms of positive numbers, a very large positive number like 300 or 400 also does not make sense, so we wanted to restrict the total marks to about hundred for at least for the Physics, Maths and Chemistry and for the total may be we want to restrict it, maybe we want to keep it to 300, something like that.

So, we could write for example the range of values are 0, 1, 2, 3, upto 100 or if you want we can write 0, 1, 2, 3 up to 300 if you want to keep up to 300 And the operations clearly what operations can be allowed on this? If you look at the operations on integers we had plus, minus, multiplication and division, we had comparison, so which of these operations are meaningful?

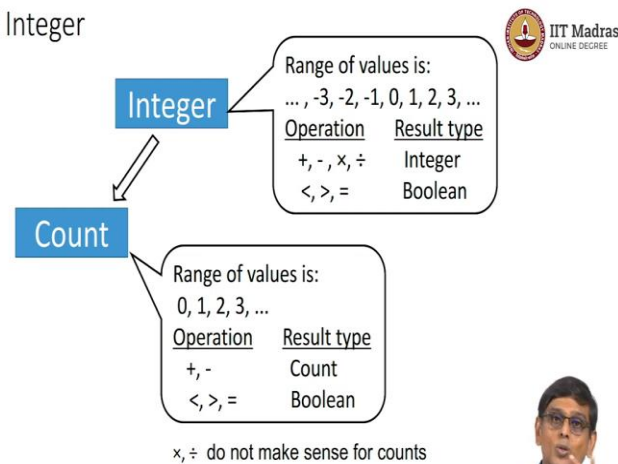
Plus clearly make sense, you can add marks to each other, in fact we did add all the Physics marks to each other to find out the total physics marks divided by the number of students to find the average physics marks and so on. So, addition of marks make sense and also you can add the physics marks, chemistry marks with the mathematics marks and get the total marks, that also you can do.

So, addition of marks make sense, subtraction also might make sense. Like for example, you might want to find the difference between the average marks and the particular marks of a student, you might want to find that or somebody has got a maximum mark, somebody has got the minimum mark, you want to find the difference between the maximum, minimum mark, so that tells you the range of marks over which students have scored, maybe the minimum marks is 40, the maximum mark is 90, students have got between 40 and 90, so 50 mark range. So, you could find difference also probably makes sense.

But definitely multiplication of marks does not make any sense and division of one mark with another mark also does not make any sense. So, we want to restrict the operations on marks to only addition and subtraction and not allow multiplication and division. Comparison of marks definitely makes sense, you can check whether two people have the same marks, you can check whether one person has less marks and other and so on, so comparison also make sense and that would return Boolean type. So, this is another subtype that we have of integers.

(Refer Slide Time: 5:14)

Subtypes of Integer



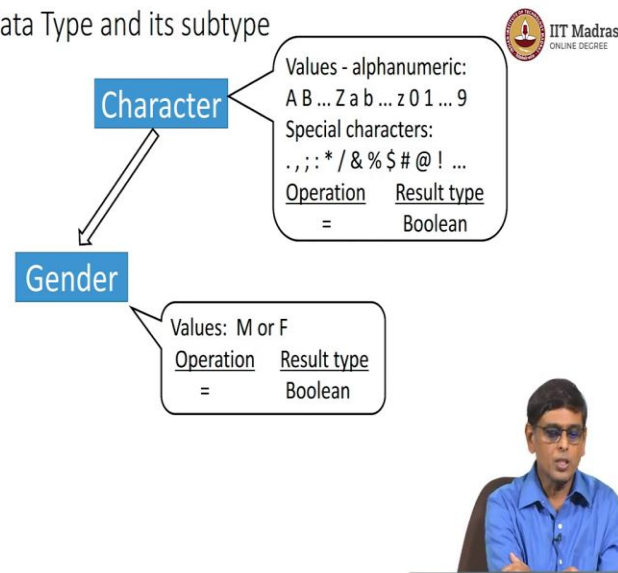
Now, the third subtype that we saw when we were doing our various things is counting we counted and we had a count variable and in that count variable we were maintaining the count of the number of cards we have seen. So, what is the range of values here again? You know basically that count values cannot be very very large because if your data set size is max count value also should be up to max only, but anyway here I have assumed basically that all positive numbers are ok, 0, 1, 2, 3 so count can be anything.

But in principle you might want to restrict it to say a max so 0, 1, 2, 3 up to a max, and what operations are allowed on this? Again for example like in marks you can add counts, I can count the males separately, I can count the females separately, the boys separately, girls separately and then I can add the two together, so two counts can be added. So, addition of count seem to make sense, similarly I can count all the boys, I can count all the girls, I can subtract, so I can check what is the difference in the size of the number of boys and number girls, I can check that by subtracting.

So, subtracting count also seems to make sense. But multiplication and division of count again does not seem to make any sense. I can also compare count, I can see whether the count of boys is less than the count of girls or whether the count of boys is equal to the count of girls, I can do that.

(Refer Slide Time: 6:32)

Character Data Type and its subtype



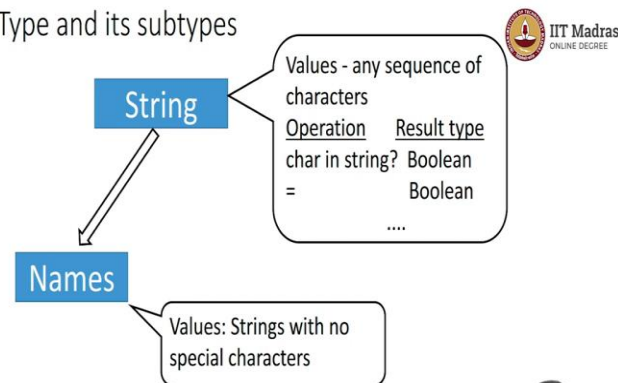
Finally, if you look at the character data type, we could basically, we have one type of field that we saw in the cards basically was the gender field. So, the gender field if we recall, had two values, the values were male or female which we denoted by M or F, the letter M or the letter F. So, basically both M and F are characters, so the gender data type is a subtype of the character data type where only two values of two character values are allowed M and F.

So, if you try to assign to a gender any character which is other than M and F. For example you try to give it A or you try to give it 1 or something like that it would give an error, it will not allow it should not allow it. So, there are only two values that are basically possible for the gender data type.

And in terms of operations you could basically check whether the gender of the card is M or not, for example is it a male or not, so equal to you could do, you could use and we have used equal-to to check whether the card has a male, M written on it or an F written on it, so that is the other, that is the operation that we can perform. So, the same operation equal to of character basically is what we say is inherited or it is basically it carries over to the subtype, gender type.

(Refer Slide Time: 7:55)

String Data Type and its subtypes



Now, we are going to introduce basically a new data type which we have used in all our cards and this data type is called string. Now, the string data type basically is something that is a sequence of characters, so its values are a sequence of characters and we do not put any restriction on the length of the sequence, so the string can be of any length.

So, it could be and we did not put any restriction also, we are not putting any restriction also on the types of characters, like the sequence of characters could be alphanumeric or it could be special characters or it could be anything, so all those characters can be there in the string. So, any sequence of any character when you put together when you make a sequence like that that thing is called a string.

And you could check, for example the operations that are possible for strings or you could check for example whether the specific character is in the string or not. Like we check for example whether there is a full stop or a semicolon or a comma or something in the string, you could check that, you could check whether string is starting with the letter A for example, you could check whether the string is ending with some letter, you could do all those things.

You could also check whether one string is equal to another string. So, to check whether one string is equal to another string you have to take the sequence of characters in the first string and take the sequence of characters in the second string and go one by one, first character, compare with the first character, first character of the first string with the first character of second string, second

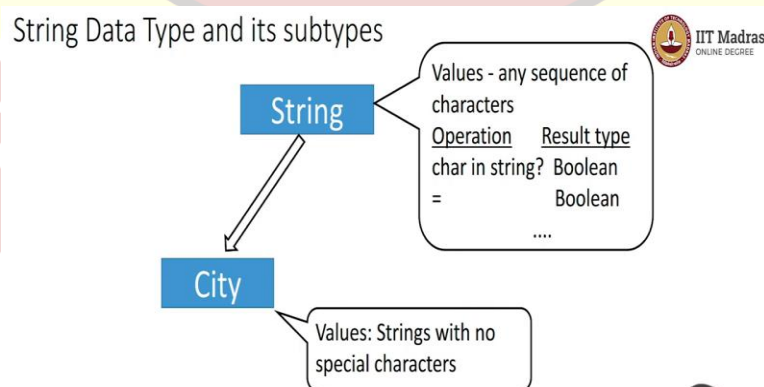
character of the first string with the second character of second string, go on like that, if all of them are equal then we say the strings are equal. So, equal to is also a valid or meaningful operation for strings and the result type of equal to would be clearly Boolean.

Now, one of the subtypes that we used in our cards basically was names. So, names are basically again strings, but there is a specific constraint that we wanted to put on names. So, here the constraint that we want to put basically is that it should not have, typically names do not have any special characters. Of course if you go to some European countries you might find the letters having some annotations on top of it, so some things like that could be there.

But at least in India we do not have any special characters in the names unless of course you want to write name like you want to write Aditi R and for the initial R you might want to put a full stop after the R, some people do that, which case you might allow the special character dot full stop in the name.

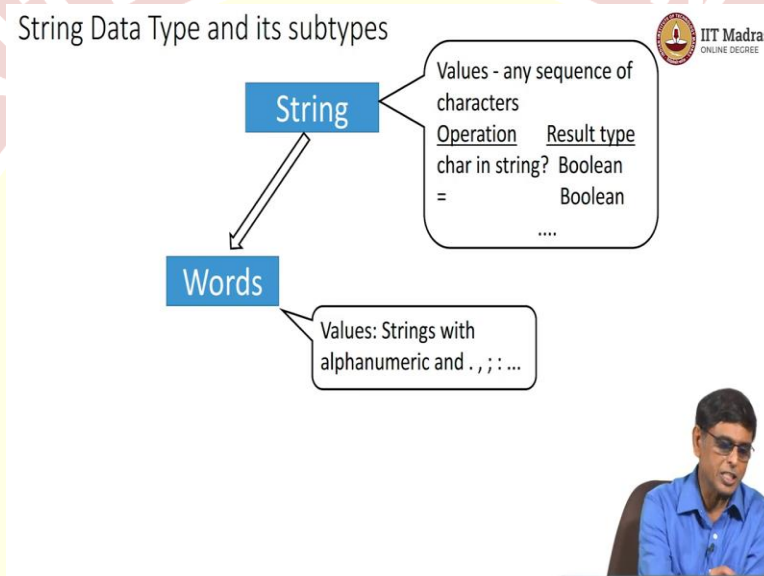
But typically names do not have special characters and they would just have either capital letters or small letters and usually would not have numbers also. So, we should have no special characters and should not have numerals and should have only the alphabet, it could be capital or small because typically you start with the capital letter for a name and then follow it up with small letters. So, you could put some constraints like that for names.

(Refer Slide Time: 10:41)



Similarly, if you take the subtype city which is another thing which we saw, we had cities like Chennai, we had Theni, we had Madurai and so on, this kind of strings again are strings with no special characters in fact again for cities it is like names one would expect basically that cities also basically do not have funny characters in them, do not have numerals in them, do not have 1, 2, 3, 4, and all in them and so on, so we should put constraints on the city name as well, so same constraints are as that we put on the name, can be put on the city subtype as well.

(Refer Slide Time: 11:18)

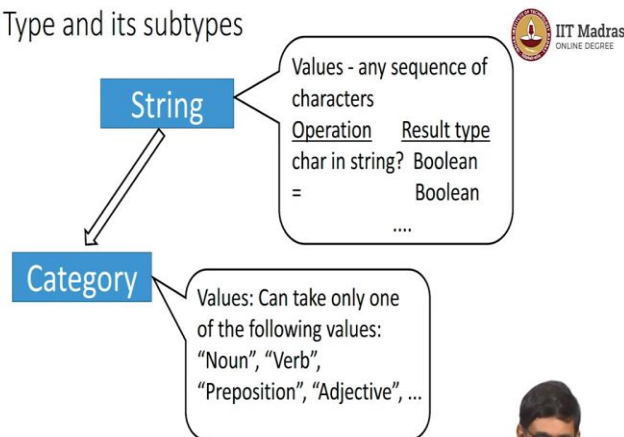


The third kind of subtype that we saw, that we used in our data set is the words that we saw in the paragraph. Like it, the, and Vedanayagam and so on, so Vedanayagam was also a name it is also a word that appeared in the set of cards. So, these words basically again are strings which can have alphanumeric characters and could also have some special characters some limited numbers of.

We had, for example at the end words we had full stop or we had a comma, or we had a colon or a semi colon, so those things are also permitted in words. And in terms of operations again the only operation that we had for string was equal to, so presumably the equal to operation we can also do for words, we can do it for names, we can do it for cities and so on, so we can check whether one city is the same as another city and so on.

(Refer Slide Time: 12:13)

String Data Type and its subtypes



Finally, we had this special thing called category in the word data set where we said basically that the category of the word could be noun, could be adjective, could be preposition and so on. So, the category data type for a word can take only finite set of values, it can take the value noun, it can take the set of verb. So, for example, Vedanayagam cannot be a category or anything else we cannot have a value for example as a category. But noun, verb, preposition, adjective, adverb, these kinds of things are okay for pronoun, these are all okay as values for category.