INDIAN INSTITUTE OF TECHNOLOGY MADRAS

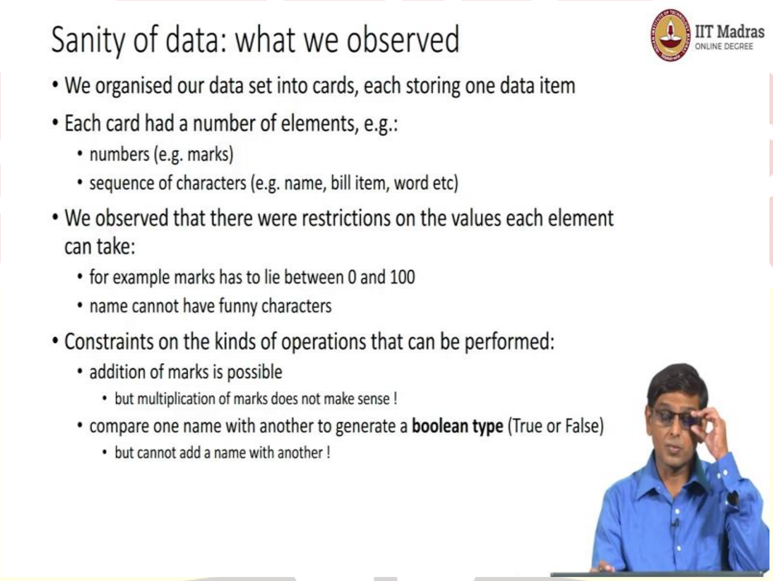
सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

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

(Refer Slide Time: 0:24)



Sanity of data: what we observed

- We organised our data set into cards, each storing one data item
- Each card had a number of elements, e.g.:
  - numbers (e.g. marks)
  - sequence of characters (e.g. name, bill item, word etc)
- We observed that there were restrictions on the values each element can take:
  - for example marks has to lie between 0 and 100
  - name cannot have funny characters
- Constraints on the kinds of operations that can be performed:
  - addition of marks is possible
    - but multiplication of marks does not make sense !
  - compare one name with another to generate a **boolean type** (True or False)
    - but cannot add a name with another !

So, in this class we are going to look at a concept called data types. So, if we look at the previous class what we did, we looked at the data sets that we have which is basically these three data sets that we have, we looked at each of the cards there and identified which of those cards had data in it which was seen, which was correct. And we identified some of the cards which had incorrect elements stored in the data items.

Like for example, we had numbers which were too large, we had category names which were wrong, we had names which were not correctly written, we had marks which were containing extra decimal places, things like that, these are incorrect. So, what we observed basically that each data item that we had stored in the card had a number of elements, these elements could be numbers, it could be a sequence of characters like a string which could represent a name or it could represent a item category or a word in the case of the paragraph and so on.

And each of these elements that are there on the card, we observed basically that there are some restrictions on the values that the element can take. For example, if you take the marks, physics marks, chemistry marks, mathematics marks or the total marks then the marks basically has to lie between 0 and 100, it cannot be a negative number, it cannot be a very large number, it cannot be like 3000 or something those numbers are not valid.

So, we have put a range, we said basically that the marks have to be in a range so let us say the marks is in the range 0 to 100. Similarly, the name cannot have funny characters, like it cannot have a semicolon or a asterisk mark or any of those exclamation mark, these are all not possible to have in the name.

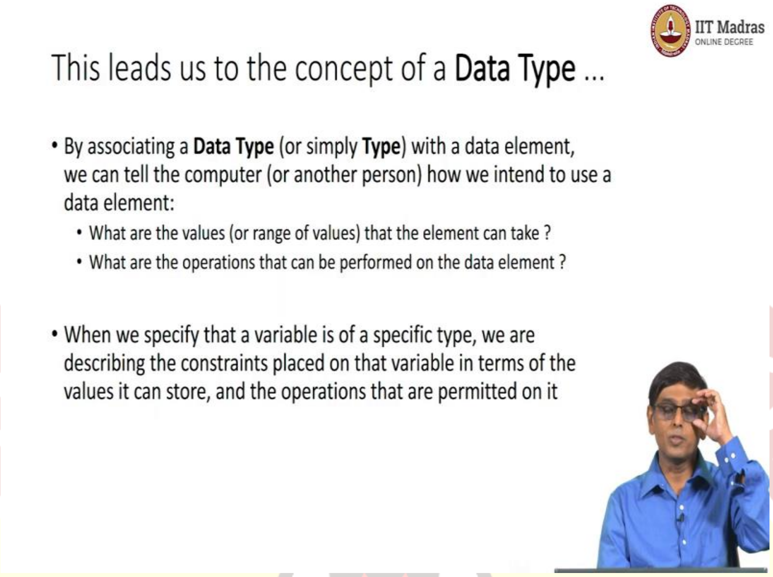
So, also we saw basically that the elements that are there in the cards, there are certain kinds of operations that we can perform on them and many operations that we cannot perform on them. For example, for marks you can add the marks together, so you can find the total of all the marks that is by adding one back to another you can keep on adding and get a total.

But it does not make sense to multiply one mark with another mark, what does that mean? I mean taking two marks and multiplying them does not mean anything at all, so multiplication does not make sense at all for marks. Similarly, if you take names, you can take two names and compare the names, you can see whether one name is the same as another name.

And that could be very simple operation, you can just check whether all the characters are the same or could be slightly more involved as we saw in the last class, where same name could be written in two different ways, you could write the first name and then the second name or you could write the last name first and then the first name second or you could just write the initial instead of the last name and then the comparison can basically check for all these conditions.

So, but there are some operations that do not make sense for names. For example, it does not make any sense to add one name with another, addition does not make sense for names. So, for every data element it looks like there is some valid set of values that that element can take which could be a range or a discrete set of values and there are certain operations that are allowed for that data element and certain operations that are not allowed for that element.

(Refer Slide Time: 3:43)



This leads us to the concept of a **Data Type** ...

- By associating a **Data Type** (or simply **Type**) with a data element, we can tell the computer (or another person) how we intend to use a data element:
  - What are the values (or range of values) that the element can take ?
  - What are the operations that can be performed on the data element ?

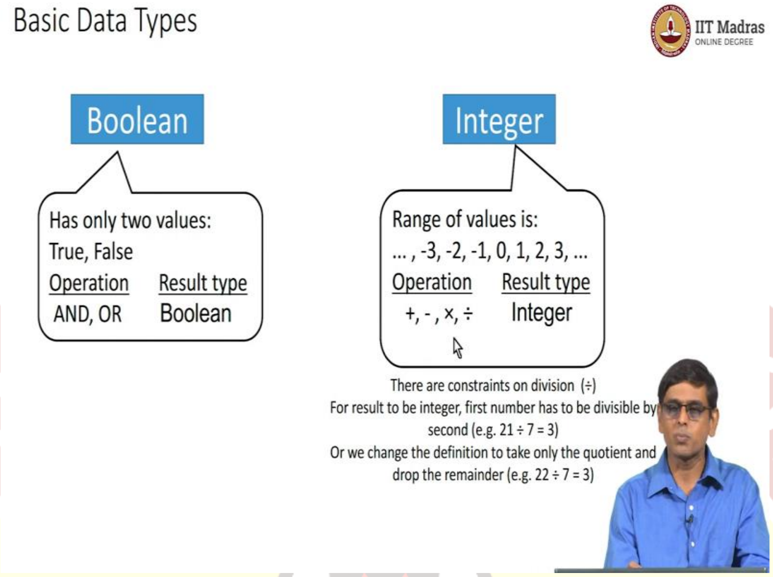
- When we specify that a variable is of a specific type, we are describing the constraints placed on that variable in terms of the values it can store, and the operations that are permitted on it

So, in computer science, in programming this basically is represented using the concept of a data type, so this leads us to this concept of data type. So, what happens in a data type basically is that data types allow us to tell the computer or maybe another person how we intend to use the data element. So, what range of values that we are allowing that element to take and what operations we want to do with that element. So, you are telling the computer constraints under which you are going to use that particular data element.

So, when we specify that a variable that we are going to use. Remember that a variable is something that we use to store intermediate values, when you say specify that a variable is of a specific type or a data type what we are saying basically is that we want the computer, if you are telling the computer we want that computer to check that the values that are being assigned to this variable are within the permitted range of values that we have specified in the data type.

And the operations that we are performing on it are also only those which are permitted. So, non permitted operations if they are found the computer should basically reject that particular operation, so that is what we expect from the data type. So, data type basically is a way to ensure some kind of sanity of the data in a more formal way.

(Refer Slide Time: 5:02)


Basic Data Types

Boolean

Has only two values:
True, False

Operation	Result type
AND, OR	Boolean

Integer

Range of values is:
… , -3, -2, -1, 0, 1, 2, 3, …

Operation	Result type
+, - , ×, ÷	Integer

There are constraints on division (÷)
For result to be integer, first number has to be divisible by second (e.g. 21 ÷ 7 = 3)
Or we change the definition to take only the quotient and drop the remainder (e.g. 22 ÷ 7 = 3)

So, let us look at some basic data types that we have already seen in our data sets. The first data type that we did not actually see in our cards but we used when we were doing our flowchart and so on is the Boolean data type. If you recall the Boolean data type is something where there are only two values for this Boolean data type, true and false.

What kind of operations can you perform on the Boolean data type? You could basically take two Boolean data types, Boolean values and you could check whether both of them are true. For example, you want to check whether or not a card is gender is male as well as whether the date of birth is within a certain range, both of those conditions evaluate to a Boolean value and you can check whether both are true by using a AND operation, so you can do an AND operation or you can do an OR operation or could there could be other operations like you could do a NOT operation and so on. These are the kind of operations that you can do on Boolean data types.

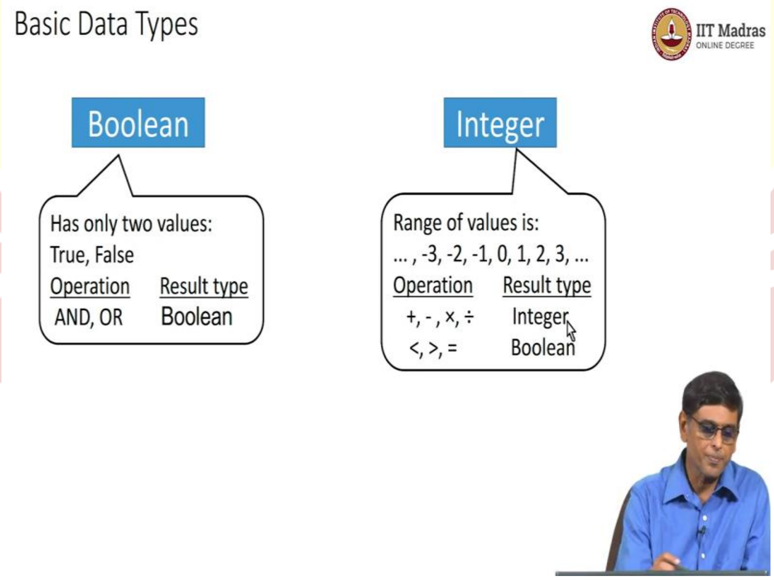
The second data type that we saw in our cards is the integer data type. Now, if you recall from our class in mathematics, integer basically is something that can take 0 or it can take positive number values or it could take negative number values. So, the numbers can be anything, any negative number minus 3, minus 2, minus 1 like that or 0 or 1, 2, 3 and it can go on like that, 4, 5, 6 and so on. So, any range of values which is basically in this negative numbers, 0 or positive numbers list.

Now, what kind of operations can we do on integers? We know from our school mathematics that what we can do with integers is we can add the integers to each other they produce integers, you can subtract integers from each other they will also produce integers, you can multiply one integer with another that also produces an integer, or you could divide integers this one is a little tricky because when you divide one integer with another it is not very clear how to do that, because if the first integer is say 21 and the second integer is say 7, then dividing 21 by 7 means something because 21 indeed is divisible by 7 and the result is 3, so 21 by 7 equal to 3 makes sense.

But if you have a number like 22 and you have the second number is 7 then 22 divided by 7 does not, 22 is not divisible by 7 or 7 does not go in 22, it is not a factor. So, if that is the case then you do not know what to do, so you need, either you have to say that the division operation is not allowed in such a case or you have to say basically that let us redefine division that when we divide 22 by 7 we take only the quotient which means that the, when you do 22 by 7 you get the quotient 3 and you get the remainder 1. So, ignore the remainder and keep only the quotient that is another way of doing it.

So, if you define it like this which is in terms of quotient, then even the division is defined now for all integers and so now you can have all the four operations plus, minus, multiplication and division for integers.

(Refer Slide Time: 8:04)



Basic Data Types

Boolean

Has only two values:
True, False

Operation	Result type
AND, OR	Boolean

Integer

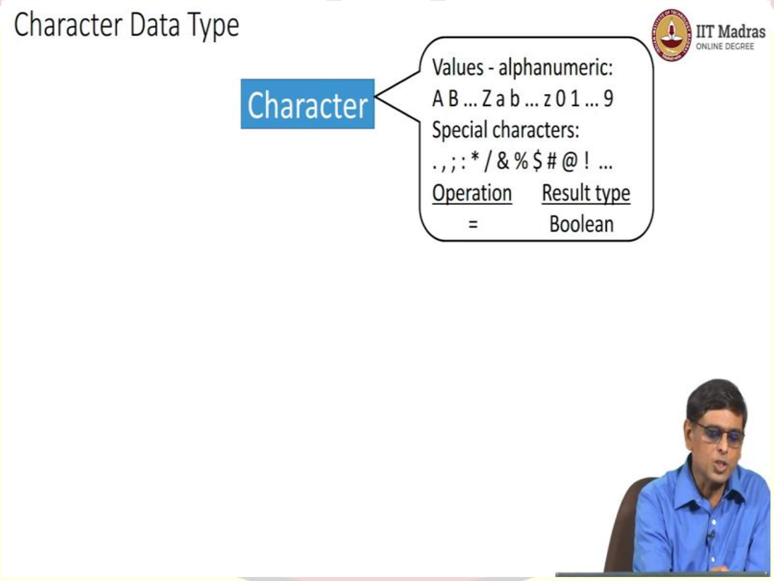
Range of values is:
... , -3, -2, -1, 0, 1, 2, 3, ...

Operation	Result type
+, - , ×, ÷	Integer
<, >, =	Boolean

The other kind of operations that we saw that we used actually for integers basically is comparison. So, we compared one integer with another to see whether it is lesser than or greater than, for example, we compared one date with another date, we compared one mark with another mark and so on, so you can compare the marks or dates or things like that, so you can check whether one is less than the other, one is greater than the other and whether one is equal to the other.

So, comparison and as we all know that comparison basically of any numbers generates true or false value, so the comparison result type is Boolean. So, any operation basically can take a single variable or can take single value or two values. So, in the case of comparison or addition or multiplication we are taking two values and performing the operation and the result can be in the case of comparison it turns out to be Boolean result, in the case of addition and so on it turns out to be an integer result.

(Refer Slide Time: 9:00)



Character Data Type

Character

Values - alphanumeric:
A B ... Z a b ... z 0 1 ... 9
Special characters:
. , ; : * / & % $ # @ ! ...

Operation	Result type
=	Boolean

So, the third data type that we have seen in our cards is the character data set, we have not really seen the character itself, you only saw the character as something that was present in the words, like we saw that some words were ending with full stop, for example, so full stop is a character. But the character data set is something where basically the values that are allowed are the alpha numeric, the standard alpha numeric values that is A to A, B, C, D and so on up to Z which could be capital letters or it could be small letters or numerals that is 0, 1, 2 to 9 so the A to Z in capitals or small along with the numeral 0 to 9, this set of characters are called the alphanumeric characters.

Besides alphanumeric characters we can also use special characters. Like for example, you can use full stop, comma, semicolon, colon, asterisk symbol, that is star or division slash and percentage and so on, these are the special characters. So, the character data set basically or data type is the set of all the alphanumeric characters and the special characters, all of them together are the characters. Those are the valid values that the character data type can take.

What operations are possible on characters? In our data set what we looked at so far, we only used the equal to operation, we wanted to check whether a given character in a word basically is a full stop, for example, so we wanted to check whether the word ends with full stop, so the last character of the word whether that is a full stop or not.

So, that means that we are checking for equality and the result of equality as we know is Boolean. So, a single operation or equality is all we needed for our data set so far, we did not use any other operation.