We're good students.

The people program with MATLAB.

So in this module today what

we are going to learn is about

local and global variables.

That is under the chapter,

function and function files.

So I'm doctor Tavares from college.

So the outline is as follows.

It gives the recover basically and recover.

What are local variables?

What are global variables?

So at the end of this module you will be

able to understand what are local variables,

what are global variables,

and understand the difference

between local and global.

So what exactly are local variables?

All we variables in a

function file are local,

so both the input and the output

arguments and any variables that

are set within a function file

are set to be a local variables.

This means that the variables

that are defined and recognized

only inside a function file.

So whenever a function file is executed,

So what MATLAB will do is it uses an

area that is separate from the workspace

which we have seen in earlier models.

That is the memory space which is

assigned whenever any variable is created.

So there is a separate work

together now in a function file.

The input variables are assigned values.

Each time the function is called.

So each time we call the function,

the values will be assigned.

These variables are then used in

calculation within any function

whatever calculation that will be done.

Now when whenever the function

file finishes execution,

the value of the output arguments get

sampled to the variables that were

used when the function was called,

so we'll see what happens

when we take a small example.

The assignment of values to these

variables in the function file will

not change assignment elsewhere.

So what what is,

is that a function file can have

variables with the same name as

variables with the command staff,

because they are going to differ at all,

so we can have the same name as

both the places in the script

file in the function file.

The both of them have different workspaces.

Local workspace for in a script

file and a separate workspace

whenever you play a function file.

A function fact doesn't recognize

variable in the sense that every

function that create a variable

which is the outside of function,

script, or maybe the command window.

So let us take an example of what

we made a local variables again.

So if you take a look over here.

We have created 2 variables that is

equal to four and be equal to 8.

So to scale.

Values are assigned to variables and B

and simple operations perform South.

South is equal to a + B that is the

sum between the two variables and D

will give us the difference between A and B.

So and we just save this simple

script and let us say right now

give it a name X sub gift order.

That is script filename and

we go to command window

and execute the command window.

So what we get is. We get South.

OK, we'll just take a look at what

is in the local workspace over here.

So all the four variables are created,

so AB is 4 BZ and is nothing

but a difference between 4:00.

That is in fact is minus four 4

- 8 - 4 and 4 + 8 gives us 12.

So this local workspace shows all the

value of the size class and so on South.

This can be easily you can access

once you learn the local workspace

we can access it in the command.

Hello, another script as well,

so these are quite a local variable.

Anything what we can see is local workspace,

a local variable.

So whenever I say yes I'll get

South is equal to the value of

easily access the variable list.

Similarly we can access DB and so on.

So these are called as local variables.

Now let us see what happened

when in a function.

Same thing as this modified the

script by and made it a function.

So the first function definition

line and the function name X subdiv.

And again,

we are now the input arguments

and nothing but output arguments.

And again the function body is

the same as is A+B is a - B.

Now we again try to run this file.

So when we try to run this function 5.

OK, again,

this value of whatever is getting

will get away and will be assigned

to sum and difference.

So we're doing it,

and to fair enough,

but now there's a difference.

Or if you can see the SNMPD which

are there within a function,

they are local to that particular

function of the local variables

of that particular function.

So now if I go to the command window,

executing the function file and I try

to say yes and try to find out a value,

I get an error and say I'm

recognized function or variable.

Is the reason being is?

There's a different community,

different workspace.

What is used when we're using functions.

So if you take a look at a workspace,

the local workspace ideas have

different sound because that was

created where command window.

So we have copied the value of.

That is visible in the local workspace the a

BSD are not visible in the local workspace,

so we can't access them,

so there is a way how we can access any

value in the function of the local variables.

OK, we'll see that that is by using

something called as global variables

or using the global keyword.

Now each function file has its

own local variables which are

not shared with other function.

Or we can workspace which we have just seen.

It is possible, however, to make a.

There was one man and several

different function files, and perhaps

in the workstation which you'll see.

This is done by declaring the variable

global and the global command has

the form of global variable.

So again variable name.

The same rules apply over here as well.

What should be the variable? No,

we can have more than one global variable.

There's no rules that one you

can have more than one,

we just have to say global.

Like I said, Global X space, Y space Z.

So this is so here we have created

three global variables XYZ.

The variable has to be declared

globally in every function that the

user wants it to be recognized there.

Then the variable is common.

In fact, only two. Both the these facts.

Next,

the variable value can be assigned

reassigned a value any of the location.

It is common.

Well,

it is normally recommended that

whenever you use the global variable,

use long descriptive names OK or

use all capital letters in order

to distinguish them from regular

or regular variables.

Then the global command must

appear before the variable is used,

because that is how we have seen

how it will be used,

and it is always recommended that

we use it at the top of the file.

We just start the function after

the function definition line.

It's better to declare it there itself,

and finally the global command has to

be also entered in the command window.

All the script file OK,

then it will be recognized in the workplace,

which we'll see shortly.

So let us take a look at this example.

Again,

it is the same thing that is

calculated something difference.

But the difference over here is you

can see that I've defined this global.

I've used a global statement global

keyword and there's space view,

so I reached small view.

But normally we try to use bigger names,

so space D it is different.

So we are declaring this S&D

to be now global variables.

Now in the command window in the

command window also we have to

suppose you want to make it common.

We again have to come to the

common window and say global space.

Now the SNMPD are no longer local variables.

You pick up your local variable from Globe.

Sorry from local they become your

global variables and then you

can see the local workspace.

The workspace earlier.

We could not see SNP could not

see SNSD now SND has come in the local space.

Once it comes in the local workspace.

That means it can be accessible both

now and it can be in the function

as well as outside of the function.

So in fact when I said now if I try

to access South I can clearly see I

can get the value of South equal to.

Read it. Whereas earlier we were

getting an error because at that time

the local workspace was completely

different from the function workspace.

In fact the variables were defined

in two different workspaces.

Now by using the Global Command,

both of them have become common and now

we can modify the values either in the

function or outside of the function as well.

Another example,

we have a small example where temperature

conversion function Celsius to function,

so again we are.

This is so this is the form normal formula

to convert from Celsius to Fahrenheit.

We get in the command window.

One way we go to call it we get it as 98.6.

We are now if we try if you see

over here if you try to access the

value of parent what is going to

happen over here is that we will get

under general function of variable.

Why because right now again this is

just local to that particular function.

Now if you want to access the value

what we can do is we can come over

here and say global Fahrenheit and come

to the command window and before we.

If you just say C227 will likely

get 98.6 but will not be able to

access the this error,

we are able to send the function for

that we have to say global Farrant.

Then if we say that we are able to create it,

we'll get the value,

which is nothing but 98.6.

Yeah, so these are the reference book.

Thank you students.