STATEMENT 1

```
from tkinter import *
```

[

Package = Collection of built in class + functions + data objects + sub-packages.

Package is stored as a .py file in Lib folder of Python installation path.

Import statement loads the package into programs memory.

If we import package then we must access every function / class in the package as follows: package_name.class_name or package_name.function_name.

To avoid it we use the altername syntax:

```
from package_name import *
```

So that all classes / functions / data objects can be access directly.

]

STATEMENT 2:

```
root_window = Tk()
```

[

Tk is a name of a built in class implemented in the tkinter package.
Putting a call operator '()' around 'Tk' we are asking Python to create an object of class Tk.

The class Tk represents 'MAIN WINDOW or ROOT WINDOW' of an application.
Therefore,

```
root_window = Tk()
```

statement creates an 'IN MEMORY COPY' of 'main window' and names it as root_window.

]

IMPORTANT PRINCIPLE:

(A) Anything that we see on screen, including text, window etc is present in program memory in the form of object of some class.

(B) If we want to chagne the properties of entities on screen such as text, window then we must make the corrossponding change by accessing 'IN MEMORY OBJECTS'

STATEMENT 3:

```
root_window.geometry('300x200')
```

Explaination of object.function() syntax.
In object oriented programming languages,

we define classes so that we can create objects from them and store appropriate data values in them.

After creating object, we want to apply functions on those objects so that some action takes place. As per rules of Object Oriented Programming, FUNCTION THAT CAN BE APPLIED MUST BE DEFINED IN THE CLASS.

Therefore, object_name.function_name(parameters) is internally converted to

class_name.function_name(object_name, parameters)

In short when we apply function to object, Python locates function in the class and sends object on which function is applied as its first parameter.

#------------------------

Applying these rules:
root_window.geometry('300x200')
is internally converted to
Tk.geometry(root_window, '300x200')
This function sets dimension of root_window to 300 pixel wide and 200 pixel in height.

STATEMENT 4:
root_window.title('My First Window')

Applying above rules the above call is converted to the following:

Tk.title(root_window, 'My First Window')

This function call sets window title to My First Window.