Ryan Snell
Programming languages
Chapter 5 Dynamic Variable

```
# program 2:   Is it possible to do in Java???
def bind_var():
   dynamic_var = "Initially a string"
   print(dynamic_var)
   dynamic_var = 123  # Rebinding with an integer
   print(dynamic_var)
bind_var()
```

Explain why, in the paragraph format, not a single sentence or two.
Note: strongly prefer posting your solutions to your github and paste the URL to the BB. If there's any write-up, either save it to a plain text file or pdf.

Python uses dynamic variables which allows you to assign any data type to a variable. Java is strictly typed and only allows one data type to be assigned to a variable. To get around this using java I changed the value to 123 as a string and assigned this to another variable using the Integer.parseInt method. Which doesn't solve the problem of assigning the different data types to the variable but allows you to separate out the integer from the string.

Before a variable can be referenced in a program, it must be bound to a data type. The two important aspects of this binding are how the type is specified and when the binding takes place. Types can be specified statically through some form of explicit or implicit declaration. With dynamic type binding, the type of a variable is not specified by a declaration statement, nor can it be determined by the spelling of its name. Instead, the variable is bound to a type when it is assigned a value in an assignment statement. When the assignment statement is executed, the variable being assigned is bound to the type of the value of the expression on the right side of the assignment. Such an assignment may also bind the variable to an address and a memory cell, because different type values may require different amounts of storage. Any variable can be assigned any type value. Furthermore, a variable's type can change any number of times during program execution. It is important to realize that the type of a variable whose type is dynamically bound may be temporary. Stack-dynamic variables are those whose storage bindings are created when their declaration statements are elaborated, but whose types are statically bound. Elaboration of such a declaration refers to the storage allocation and binding process indicated by the declaration, which takes place when execution reaches the code to which the declaration is attached. Therefore, elaboration occurs during run time. For example, the variable declarations that appear at the beginning of a Java method are elaborated when the method is called and the variables defined by those declarations are deallocated when the method completes its execution. As their name indicates, stack-dynamic variables are allocated from the run-time stack.