CSE324 Quiz #7 Name:_____**KEY**_____

True or False: (all questions are about **Ada**), one point for very True/False question. True's are marked T

a) In general**, dynamic** and **static genericity** are of equal **polymorphic *power*** of code sharing. **F**
**Dynamic genericity facilitates machine code sharing, e.g., ADT's set of operations, for much larger instances of the subject generic module (for different types' elements) than the static implementation. In case of the static genericity we must write a separate module for every type element resulting in much more none shared operations' code at run time.**

b) ***Unconstrained*** arrays and ***discriminant*** (parameterized) records in Ada have **dynamic *polymorphic*** power. **F**
Different instances of both are to be instantiated at **compile time**, yielding ***many*** instances.

c) Ada generic packages' mechanism provides ***static*** **genericity.** T
Generic package instantiation is 100% a compile time issue, hence **static**!

d) **Generic packages** are used for better **static** typing **security** and **faster** run time code execution. **F**
**Generic packages' facility is mainly for saving the programmers writing of packages' codes, no effect on any typing security or speed of code execution run time.**

e) Ada **pointers** are **typed** for better language security. T

f) **Name access** (visibility) in Ada's internal packages is **identical** to that of Modula-2. **F**
**There is no internal package in Ada with such sophisticated name visibility as in Modula-2 internal modules. We did not talk about any similarity in the class!**

g) ***Ada*** is less secure than MODULA-2. **F**

**NOP! It has static type checking without dynamic genericity, secure discriminant (parameterized) records.**

h) Ada is well suited for military applications, much better than Pascal and Modula-2, since its translated code runs much faster. **F**

**Although its suited for military applications but not because its code runs faster, instead it has exception mechanism.**

i) The **explicit** declared type of any Ada's constant "**named number"** is LONG_INTEGER. **F**

**Not "any" constant named type, we might declare INTEGER, REAL, CHAR also. But, the idea is to leave it to be implicitly explored by the compiler anonymous type, for better portability.**

j) **"null"** is a valid statement that does nothing (no action), in Ada. **T**

k) Ada is more portable than Pascal & Modula-2. **T**

l) Ada has a richer set of types than Pascal & Modula-2. **T**

m) Ada arrays and records are more powerful than those of Pascal and Modula-2. **T**

n) The Ada typing system is a tradeoff between gaining power and losing run time efficiency, e.g., many possibilities of "constraint" run time checks, due the utilization of such power. **T**

o) Ada has “opaque” and “open” types same as in Modula-2. **F “open” is static not as the Modula-2 dynamic.**