

1) True/False: (10 pts)

- a) In the **pure** object oriented domain (POOD) Every datum in the language is an Object. **T**
- b) An object in the POOD is a **true** Abstract Data Type (ADT). **T**
- c) In general, an object in the POOD a "*state*" and a "*behavior*". **T**
- d) We can safely say that the POOD is a message based system, where data objects exchange messages to act on themselves and possibly on others. **T**
- e) The OOD's HLLs is less secure than the block structured HLLs since there is no explicit typing system and the associated type checking mechanism. **F**
(**type checking is replaced by protocol/behavior checking**)
- f) A class in the OOD HLLs is the abstraction of all similar behavior and state objects. **F (similar behavior only not state)**
- g) A **class instance** will have additional behavior (methods) than its instantiating class, just in case it does some extra unique activities. **F**
(**NOP! No more than the list of instance methods in the class!**)
- h) In POOD a **class** has its own behavior&state separate from its instances. **T**
- i) The *state* and the *behavior* of an *instance* of some class C is the addition of **all** "*states*"(variables) and "*behaviors*"(methods) in C and its ancestors until the root "**Object**" class in the **inheritance** hierarchy's tree. **F**
(**NOP! Not all states and behaviors, only instances' variables & methods**)
- j) The concept of data as objects presents some **implicit equivalency** between *code* and *data*, in the OOD HLLs. **T (data behave/act as a code does)**

2) In the following Smalltalk code:

i) `x := 'Soccer man'. x at:9 put:$f.`

(5pts)

What is the:

- a) the second message recipient object and its class-type? **x , string**
- b) the message name and type? **at:put: , keyword**
- c) the new **state (value)** of the recipient object after executing the message's method? **'Soccer fan'**

ii) `sum := 1. 1 to: 11 by: 7 do:[x| sum := sum *x]. sum.`

(5pts)

- a) what is the output of executing the above code? **8**
- b) What is the keyword message used in the above code? **to:by:do:**

c) Identify the classes of recipient of the keyword message, in part b above.

Number