**In-class Group Exercises**

**Handling Arrays, Objects, and OOP Features during Code Generation**

**Why?**

**Learning Objectives:**

1. Familiarize yourself with the requirements for PP5 for OOP
2. Familiarize yourself with array layout and manipulation
3. Understand what constitutes an object’s memory and vtables
4. Understand what TAC code is generated to implement a polymorphic call site versus a static call site, using the stack, object memory layout and vtables

**Resources:**

1. PP5 assignment sheet
2. PP5 starter files
3. Handout with t3.decaf, t3.tac, t7.decaf, t7.tac, t8.decaf, t8.tac

**Tasks:**

1. In your group, assign new team roles:
	1. **Manager** - ensuring that everyone is fulfilling their roles, that assigned tasks are being accomplished and that all members of the team are participating in the activity? This is the only person that may ask questions or have contact with the facilitator.
	2. **Reader** - responsible for reading out loud so everyone on the team stays together for the activity
	3. **Recorder** - responsible for writing down important ideas/concepts and ensuring that the rest of the team is completing the answers on their activity sheet
	4. **Presenter** - responsible for reporting to the facilitator and the rest of the class when asked
2. Using the handout with t3.decaf and t3.tac, draw lines to segment which instructions in t3.tac correspond to different instructions in t3.decaf and label each segment in the TAC with the Decaf instruction.
3. As a group, answer the following key questions with respect to t3. The recorder should type the answers in a file that can be shared with the whole group and used by the Presenter to explain to the class.
	1. What constitutes the 12 in \_tmp0 = 12?
	2. Draw a picture of the stack and betsy object after the betsy.Init(100,122) returns, indicating which fields are located at which offsets and other contents as well as values.
	3. What constitutes the 40 in BeginFunc 40 ; ?
	4. Draw a picture of the runtime stack just before PopParams 12; on the first column of the handout.
4. Using the handout with t7 and t8 Decaf and generated TAC, draw lines to segment and label the TAC with respect to the Decaf instructions.
5. As a group, answer the following key questions:
	1. For t7, draw the memory for objects betsy and b, and the vtables for each class.
	2. For t7, draw the activation record for an activation of InitCow.
	3. For t7, have one person explain what the TAC code is doing for the call betsy.InitCow and b.GetMom. Be sure everyone follows the explanation.
	4. For t8, draw the memory for object veggies after New(vegetable) call.
	5. For t8, draw the vtables for each class Squash, Vegetable, Seeds.
	6. For t8, draw the activation runtime stack when veg.Grow is executing.
	7. Have one person explain the TAC and Decaf code for the calls veggies[1].Eat(veggies[0]) and call veg.Grow(s,w). Be sure that everyone understands how the vtables and objects and stack are being used and manipulated for these calls.