1. Briefly summarize in your own words the different implementation models of parameter passing discussed in section 9.5.2.

2. The textbook describes function pointers in C++ as a mechanism for calling subprograms indirectly. Write a simple C++ program that defines two different functions (f and g) that both take two int values and return an int value. Also define a function called “higher_order” that takes a function as a parameter (via a function pointer with the same signature as f and g) and two int values, and then calls the passed in function over the two values, returning the result. Write a main function that outputs the result of calling the “higher order” function twice: once passing in f and once passing in g.

3. In your own words, describe the three choices of referencing environments given in the book for passing subprograms to functions in languages that support nested subprograms.

4. Explain in your own words what the text means when it says “a closure is a subprogram and the referencing environment where it was defined.” In particular, why does it mention the referencing environment here?