

PPL Answers for part 1:

1. Dimensions of variability across programming paradigms:
 - **Code Organization:** defines how code is organized and divided to hierarchy of units in the language. Example of units are expressions , functions, classes , modules , and packages . Different paradigms have different units that build the hierarchy of the languages.
 - **Testing:** how easy it is to test and verify that code is correct.
Different paradigms have different structures and this influence the way tests are conducted in each paradigm.
 - **Syntax:** how natural readable and concise is the expression of code given the syntax of the language . Can the syntax of language be extended by the programmer (for example by adding special Key-Words , or new structures.)
2. ***(a) (x, y) => x + y**
Type: (x: number , y:number) => number
***(b) x => x[0]**
Type: <T>(x : T[])=>T
***(c) (x, y) => x ? y : -y**
Type: (x: boolean , y: number)=> number
3. 'Shortcut Semantics' is a concept that infers to an implementation of a function. It means that if an output of a function can be calculated in few ways (and receiving the same output with the same inputs) , 'Shortcut Semantic' implementation stops the calculation once it achieved the right output even in the middle of the calculation , whether 'Non – Shortcut Semantics' implementation would finish the calculation and only then return the answer.

An example for 'Shortcut Semantic' is the implementation of 'some' and 'every' functions we saw in practical session. 'some' returns true if an array has one element that satisfy the predicate. We can implement this function by iterating over every predicate of an array and return 'true' if at the end of the calculation we found at least one element that satisfies the predicate. Or we can use 'Shortcut Semantics' in our implementation and stop the calculation once we found the element that satisfies the predicate without checking the rest of the elements.