Name: Brihat Ratna Bajracharya

Roll No.: 19/075

Assignment #2 (Principles of Programming Language)

Question

What is orthogonality property. Consider C Programming language. Does it have orthogonality property or not, explain with examples.

Answer

Orthogonality Property in Programming Language

The word orthogonal comes from the mathematical concept of orthogonal vectors, which are independent of each other. In programming language, orthogonality means a set of few basic constructs can be combined in a finite set of ways to build the control and data structures of the language. Thus formed combination of construct should be legal and meaningful. Also, the meaning of an orthogonal language is independent of the context of its appearance in a program. The term is most-frequently used regarding assembly instruction sets, as orthogonal instruction set. This property is associated with simplicity i.e. orthogonality follows from the symmetry of relationships among primitive constructs, in which the more orthogonal the design, the fewer exceptions in the programming language construct. Fewer exceptions mean a higher degree of regularity in the design, which makes the language easier to learn, read, and implement in programs. So, a programming language can be said to follow orthogonality property if,

1. the language has a finite set of primitive constructs that can be combined in finite ways to form other data and control structures, and
2. thus formed structures are legal and meaningful.

C Programming Language and Orthogonality

C Programming Language lacks the orthogonality principle as we can find many asymmetries in the language constructs. We can begin with the void data type. If any function does not have a return type or does not return anything we use void in function definition as,

```
void some_function(arg1, arg,2) {} // void return typw
```

but there is no provision of declaring a void variable as

```
void some_variable; // this is not allowed
```

Another inconsistency is with two kind of structured data structure: arrays and struct (record). We cannot return entire array in a function but we can return struct data type. With arrays, we can return pointer to the array by specifying array's name without index. But if arrays are inside struct, we can return the array wrapped inside struct.

Also all other data types are passed as parameters as a value but arrays are passed by reference i.e. the memory location of first element of array is passed and subsequent elements of array are accessed by increasing the index of the array.

All these exceptions leads C Programming language to be non-orthogonal language.