Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)\

**Metropolitan State University**

**ICS-365-01 —Organization of Programming Languages**

**Homework #1**

\* Due Date – See Class Schedule \*

\* Use your own words to answer the questions, do not cut and paste from another source, that is cheating \*

\* Also, your answers on Q8 and 10, need to be thoughtful, not just one sentence

1. What would be the most likely cause as to the most common syntax error in LISP programs.

Answer: I think the most likely cause to common syntax error in LISP programs language should be the confusion or forgetful to close or open the parentheses or round brackets when writing a long code line or multiple line.

1. How were the pseudocodes of the early 1950’s implemented?

Answer: The pseudocodes implemented as very poor, and difficilt to implement because in that time the computers were far less usable, being slow, unreliable, expensive, and having very small memories and difficult to program. For example, an ADD instruction might be specified by the 14 rather than a connotative textual name, even if only a single letter. So it makes programs very difficult to read. On another hand, Short Code which was developed by John Mauchly in 1949 for the BINAC computer, was one of the first successful stored-program electronic computer. But Short Code was not translated to machine code, because it was implemented with a pure interpreter called automatic programming. It was 50 times slower than machine code. Another one is Speedcoding (Backus, 1954), beign developed that extended machine languages to include floating-point operations. The Speedcoding could be programmed faster than machine code. The last one is the UNIVAC “Compiling” System(1951-1953, team led by Grace Hopper). It expanded a pseudocode into machine code subprograms in the same way as macros are expended into assembly language.

1. What version of Fortran was the first to have any sort of dynamic variables/

Answer: Fortran 90 (ANSI, 1992)

1. What are some features of a specific programming language you know whose rationales are a mystery to you?

Answer: The features of programming language that mystery to me are some features in Javascript like “Call back function” in Node.js, framework of Javascript. There also has so many syntactical problems that it’s positively dangerous to use because JavaScript can actually fail silently due to syntactical errors that means no error messages to warning.

1. What programming language dominated
   1. Scientific computing?

Answer: **Fortran** has dominated scientific computing over past 50 years.

* 1. Business computing?

Answer: **COBOL** has dominated business application over past 50 years.

* 1. Artificial intelligence?

Answer: the functional language **LISP** has dominated over the past 50 year.

* 1. Operating systems?

Answer: **C** programming language dominated nowadays Operating Systems like UNIX, Linux, Windows, etc.

1. What construct of a programming language provides process abstraction?

Answer: Primitive, a simple example of process abstraction is the use of a **subprogram** that used to implement a sort algorithm that is required several times in a program.

1. What are the 3 fundamental features of object-oriented programming? What language was the first to support them?

Answer: The 3 fundamental features of OOP are **Encapsulation**, **Inheritance**, and **Polymorphism**. The first to support them is **SmallTalk** programming language

1. Why, **in your opinion (do not cut and paste an answer)**, do scripting languages appear more frequently than new compiled langagues?

Answer: Because they are smaller, simpler and focused on more narrow applications that means their libraries need not be large or big. On another hand, they easy to learn quickly and readable and easy to debug.

1. What other logic programming languages exist beyond prolog?

Answer: Prolog++ supports OOP; Abuductive logic programming extends of normal logic programming; Metalogic programming; Contraint login programming, combines Horn clause login programming with constraint solving.

1. Describe in **your own words (do not cut and paste an answer)** the concept of orthogonality in programming language design?

Answer: orthogonality means that being able to change one thing but without effect on another part. Orthogonality in the programming language design means that when you execute an instruction, just only that instruction happens, other happen nothing and having side effects.