C+ or See Positive, a new typed, imperative language for optimism.

Dr Bheemaiah, Anil Kumar, A.B Seattle W.A 98125
miyawaki@yopmail.com

Abstract:

Keywords:

What:

C+ or See Positive, is an object oriented language, different from C++, it is inspired by psychology and H.H. Dalai Lama and Bishop Tutu's philosophies of positification() and optimism(),(Lama, Tutu, and Abrams 2016) much to increase the happiness index.

Semantic transfer or ST is central to C+, much like the structures and unions of C, and pointer arithmetic, C+ moves away like Java, to value addressing, cloud functions and a thin implementation.

Reflection(Brose 1999; Morris, n.d.; Forman and Forman 2004) is not added, as the compiler is built on top of gcc, much like C++ was. C+ creates object functionality with encapsulation, inheritance and overloading all built into the ST mechanisms.

Inheritance is much stronger in the ST system with direct inheritance from a library, with transparent linking, streams(Langer and Kreft 2000) are a central part of C+ , much like C(Kernighan and Ritchie 1988; Kernighan 2005), with a unified I/O and program space.

Code reuse is central to C+ as it was invented to integrate legacy C code with newer Java and C# code, code search engines are built into the C+ specification leading to code capture for green functions, cloud functions with recycled code(Gallardo-Valencia and Sim 2014).

positification() and optimism() are central to C+ coding defined as operators in an operator library, With a JSON representation of speech as input, operators act on JSON for various 'effects', including side-effects.

Code generators are central to the design of the C+ language as part of the ST mechanism, this implements much of the research by Google, Microsoft, Amazon, Facebook, Netflix and other corporate research sources, coupled with academic research in code generation.

How:

Formal specifications for the C+ language are described with the definition of the ST mechanisms, library structure, type definitions, scoping, data structures as collections, streams and the search engine for cloud function generation.

Why:

Objective C was an object oriented language predating Java, but without semantic transfer capabilities, C+ is a code generation and code recycling language, based on cloud resident code and libraries, for easy linking, it is a next generation language for robotics, where C and C++ code dominates. C+ is a also a code language for psychology engineering, as applied in avatar therapy, in template design for AWS Lex based conversational UI.

Summary:

Main Points:
Formal Specifications:
Thread and block declarations.
ST or Semantic Transfer.
Data Structures and Collections.
Cloud Functions.
Green Function Generator.

Applications:
Directly applied to template based code generation, illustrated in an accompanying paper on computer assisted therapies of Avatar Therapy.

Code Base:

Introduction.

Problem Definition.

Background.

<original-contribution>

Formal Definitions:

</original-contribution>

Discussion.

Future Work.

References.

Brose, Gerald. 1999. “Reflection in Java, CORBA Und JacORB.” *JIT’98 Java-Informations-Tage 1998*. https://doi.org/10.1007/978-3-642-59984-2_21.

Forman, Ira R., and Nate Forman. 2004. *Java Reflection in Action*. Manning Publications

Company.

Gallardo-Valencia, Rosalva E., and Susan Elliott Sim. 2014. *Source Code Seeking on the Web: A Survey of Empirical Studies and Tools*. Lulu.com.

Kernighan, Brian W. 2005. *C Programming Language (2E)*.

Kernighan, Brian W., and Dennis Ritchie. 1988. *C Programming Language*. Prentice Hall.

Lama, Dalai, Desmond Tutu, and Douglas Carlton Abrams. 2016. *The Book of Joy: Lasting Happiness in a Changing World*. Penguin.

Langer, Angelika, and Klaus Kreft. 2000. *Standard C++ IOStreams and Locales: Advanced Programmer's Guide and Reference*. Addison-Wesley Professional.

Morris, D. S. n.d. "Automatically Grading Java Programming Assignments via Reflection, Inheritance, and Regular Expressions." *32nd Annual Frontiers in Education*. https://doi.org/10.1109/fie.2002.1157985.