BuckCal

for when you need something smarter to Calculate your Bucks ($$$), amongst other things
The Team

Ahmad Maruf

Lan Yang

Lingyuan He

Meng Wang

Prachi Shukla
The Overview

• BuckCal is a matrix manipulation language
• Essentially a scripting-style language with full support for mathematical matrix operation and spreadsheet calculation
The Structure

1. Lexical Analysis
2. Parsing
3. Semantic Check
4. AST Conversion
5. Code Generation
The Implementation in OCaml and C++

Matrix literal

BuckCal

disp {1, 2, x};

C++

int T_0[] = {1, 2, x};
int mat TT_1 = int_mat(T_0, 1, 3);
cout<< TT_1 << endl;
Implementation in C++

• Why C++?
  – One abstraction level down
  – Typed with primitives similar to BuckCal
  – Class abstraction with operator overload to easily add operation
Implementation in C++

• Structure:
  – Class int_mat / double_mat / string_mat;
  – Operators (+ - * /) in between int/int_mat double/double_mat
  – Operator << for matrix output
  – Built-in functions concerning concatenation, row/col names and conversion
Testing

```
translate
sample.bc

compile
sample.cpp

run
sample.bin

sample.out

bad case: 2 >> sample.out

0: Success
1: fail

good case: 1 >> sample.out

diff

ideal.out
```

```
2 > sample.out

2 >> sample.out
```
Combat Issues

- **Reminder of bad samples**
  - #65 opened 3 days ago by lingyuan-he
  - Status: enhancement

- **Import buckcal_lib.bc from a imported file.**
  - #64 opened 3 days ago by lingyuan-he
  - Status: bug

- **Test Sample Error**
  - #63 opened 3 days ago by lingyuan-he

- **Still problem with function overloading**
  - #60 opened 5 days ago by Lan-Yang

- **Built in function error**
  - #59 opened 5 days ago by Lan-Yang
The Lessons Learned

• Delivering good work is not dictatorship
• It’s more about making a timeline
• A complete testing framework should be built ASAP
• Get people to work in small teams
• Keep the modules small
The Demos