BURGer

PLT FALL 2017

TEAM MEMBERS

Jordan

Manager

Jacqueline

Language Guru

Adrian

System Architect

Ashley

Tester

ft. **Frederick (“Freddy”) Kellison-Linn**

OVERVIEW

- General-purpose programming language
- BURGer is statically typed and convenient to use
- Allows top-level code
- Functions

ARCHITECTURE

SCANNER

PARSER

SEMANTIC CHECKING

CODE GENERATION

LLVM IR

BURGer EXECUTABLE

TYPES

- BURGer supports integers, booleans, strings, and a null type
- BURGer has explicit type declarations
- BURGer supports variable initialization

```
int x;
int y = 12;
bool z = true;
string hello = “Hello World!”;
```

SYNTAX

Control Flow

```
int i;
i = 0;

while (i < 5)
{
   println(”yum");
   println(”burger");
   i = i+1;
}
```

Function Declaration

```
def int add(int a, int b)
{
   println(“add!”)
   return a + b;
}


print(add(1, 3))
```

BEHIND THE SCENES

- A ”hidden” main function wraps top-level BURGer code
- Print functions are implemented in a standard library written in C

PROCESS

- Group meetings on Monday, meetings with Freddy on Tuesdays

LESSONS LEARNED

- Functional programming
- How a compiler works
- Start early
- Use resources available
- Be realistic

DEMO