Compilers-I
CS 3510 Spring 2019
Programming Assignment 0:
Toy Cool Programs

INSTRUCTOR: DR. Ramakrishna Upadrasta

By:
Vijay Tadikamalla
CS17BTECH11040

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Incorrect Programs

Program1

The Identifier begins with capital letter i.e Str instead if str

Error shown:

"incorrect1.cl", line 2: syntax error at or near TYPEID = Str
"incorrect1.cl", line 4: syntax error at or near TYPEID = Str
Compilation halted due to lex and parse errors

Program2

Unterminated string constant. It can be avoid with a backslash

Error shown:

"incorrect2.cl", line 3: syntax error at or near ERROR = Unterminated string constant
Compilation halted due to lex and parse errors

Program3

One dash is missing

Error shown:

"incorrect3.cl", line 4: syntax error at or near '-'
Compilation halted due to lex and parse errors

Program4

True is case sensitive keyword.

Error shown:

"incorrect4.cl", line 2: syntax error at or near TYPEID = True
Compilation halted due to lex and parse errors

Program5

Zero Width Space is not recognized by cool

Error Shown:

"incorrect5.cl", line 5: syntax error at or near ERROR = \342
Compilation halted due to lex and parse errors

Correct Cool programs

Some common Mips instructions

- addiu → Add immediate unsigned (no overflow)
- sw → Store word
- move → move value stored at address to registers
- bne → Branch on not equal
- class_nameTab section contains information about the classes like str_const parts
- str_const(I) sections contains all the string literals in our code section.
- Function calls corresponds to using branch instructions in the assembly

Program1:Factorial

```
Main.factorial:
        addiu   $sp $sp -20
        sw      $fp 20($sp)
        sw      $s0 16($sp)
        sw      $ra 12($sp)
        addiu   $fp $sp 4
        move    $s0 $a0
        lw      $s1 20($fp)
        la      $t2 int_const0
        move    $t1 $s1
        la      $a0 bool_const1
        beq     $t1 $t2 label2
        la      $a1 bool_const0
        jal     equality_test
```

Program2:Fibonacci

```
Main.fibonacci:
        addiu   $sp $sp -16
        sw      $fp 16($sp)
        sw      $s0 12($sp)
        sw      $ra 8($sp)
        addiu   $fp $sp 4
```

```
    move    $s0 $a0
    lw      $s1 16($fp)
    la      $t2 int_const1
    move    $t1 $s1
    la      $a0 bool_const1
    beq     $t1 $t2 label2
    la      $a1 bool_const0
    jal     equality_test
```

Program3: palindrome

```
Main.check_palindrome:
    addiu   $sp $sp -16
    sw      $fp 16($sp)
    sw      $s0 12($sp)
    sw      $ra 8($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
    lw      $a0 16($fp)
    bne     $a0 $zero label3
    la      $a0 str_const3
    li      $t1 1
    jal     _dispatch_abort
```

Program4: Sum

```
Main.sum:
    addiu   $sp $sp -20
    sw      $fp 20($sp)
    sw      $s0 16($sp)
    sw      $ra 12($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
    lw      $s1 20($fp)
    lw      $s2 20($fp)
    la      $a0 int_const1
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s2)
    add     $t1 $t1 $t2
```

```
    sw      $t1 12($a0)
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s1)
    mul     $t1 $t1 $t2
    sw      $t1 12($a0)
    move    $s1 $a0
    la      $a0 int_const2
    jal     Object.copy
    lw      $t2 12($a0)
    lw      $t1 12($s1)
    div     $t1 $t1 $t2
    sw      $t1 12($a0)
    lw      $fp 20($sp)
    lw      $s0 16($sp)
    lw      $ra 12($sp)
    addiu   $sp $sp 24
    jr      $ra
```

Program5: Pattern

```
Main.pattern:
    addiu   $sp $sp -16
    sw      $fp 16($sp)
    sw      $s0 12($sp)
    sw      $ra 8($sp)
    addiu   $fp $sp 4
    move    $s0 $a0
```