

Compilers Assignment-1

COOL PROGRAMS

CS18BTECH11018

INTRODUCTION

This report has analysis of twelve(12) programs written in COOL programming language. Seven(7) out of the twelve(12) are correct programs and are analyzed using their MIPS code. There are five(5) incorrect programs and error in each program is explained in this report.

CORRECT PROGRAMS:(1-5:Trivial, 6-7:Non-Trivial)

MIPS Code of part of the program which does major computation is attached for each correct program.

Some common Mips instructions

- 1.addiu → Add immediate unsigned
2. sw → Store word
3. move → move value stored at address to registers
4. bne → Branch on not equal

Main.modf:

```
addiu  $sp $sp -20
```

```
sw     $fp 20($sp)
```

```
sw     $s0 16($sp)
```

```
sw     $ra 12($sp)
```

```

addiu  $fp $sp 4
move   $s0 $a0
lw     $s1 24($fp)
lw     $s2 24($fp)
lw     $a0 20($fp)
jal    Object.copy
lw     $t2 12($a0)
lw     $t1 12($s2)
div    $t1 $t1 $t2
sw     $t1 12($a0)
move   $s2 $a0
lw     $a0 20($fp)
jal    Object.copy
lw     $t2 12($a0)
lw     $t1 12($s2)
mul    $t1 $t1 $t2
sw     $t1 12($a0)
jal    Object.copy
lw     $t2 12($a0)
lw     $t1 12($s1)
sub    $t1 $t1 $t2
sw     $t1 12($a0)
lw     $fp 20($sp)
lw     $s0 16($sp)

```

```
lw  $ra 12($sp)

addiu  $sp $sp 28

jr  $ra
```

Main.modf: It is the modulus function used in most of 7 correct programs, so i have just attached it only once to keep the report short.

1.factors.cl(Find factors of given number)

Main.main:

```
addiu  $sp $sp -16

sw  $fp 16($sp)

sw  $s0 12($sp)

sw  $ra 8($sp)

addiu  $fp $sp 4

move  $s0 $a0

la  $a0 str_const0

sw  $a0 0($sp)

addiu  $sp $sp -4

move  $a0 $s0

bne  $a0 $zero label0

la  $a0 str_const5

li  $t1 1

jal  _dispatch_abort
```

2. gcd.cl(Calculates the gcd of entered numbers):

Main.gcdf:

```
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_const0
move   $t1 $s1
la     $a0 bool_const1
beq    $t1 $t2 label2
la     $a1 bool_const0
jal    equality_test
```

3.maxelement.cl(Calculates the maximum of entered N numbers):

Main.main:

```
addiu  $sp $sp -20
sw     $fp 20($sp)
sw     $s0 16($sp)
sw     $ra 12($sp)
addiu  $fp $sp 4
```

```

move    $s0 $a0

la      $a0 str_const0

sw      $a0 0($sp)

addiu   $sp $sp -4

move    $a0 $s0

bne     $a0 $zero label0

la      $a0 str_const9

li      $t1 1

jal     _dispatch_abort

```

4. power.cl(Calculates a^b):

Main.powers:

```

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
```

5. sumsquaresn.cl(Calculates sum of squares of 1st n natural numbers(recursively)):

Main.sumnsq:

```
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_const1
move $t1 $s1
la $a0 bool_const1
beq $t1 $t2 label2
la $a1 bool_const0
jal equality_tes
```

6. matrixarith.cl(Calculates addition/subtraction/multiplication of 3*3 matrices):

Caution : Please enter one element at a time.

First choose the operation and enter elements of two matrices(one element at once).

Main.main:

```
addiu $sp $sp -20
sw $fp 20($sp)
```

```

sw    $s0 16($sp)

sw    $ra 12($sp)

addiu $fp $sp 4

move  $s0 $a0

la    $a0 str_const0

sw    $a0 0($sp)

addiu $sp $sp -4

move  $a0 $s0

bne   $a0 $zero label0

la    $a0 str_const10

li    $t1 1

jal   _dispatch_abort

```

7. numsysconversion.cl(Converts decimal number to binary/hexa-decimal or vice versa):

Choose 1 out of 4 operations specified and enter valid input to get a number in required number system.

Main.binary:

```

addiu $sp $sp -20

sw    $fp 20($sp)

sw    $s0 16($sp)

sw    $ra 12($sp)

addiu $fp $sp 4

move  $s0 $a0

```

Main.decimal:

```
addiu  $sp $sp -20
sw     $fp 20($sp)
sw     $s0 16($sp)
sw     $ra 12($sp)
addiu  $fp $sp 4
move   $s0 $a0
```

Main.hexa:

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

Main.hexadeci:

```
addiu  $sp $sp -20
sw     $fp 20($sp)
sw     $s0 16($sp)
sw     $ra 12($sp)
addiu  $fp $sp 4
move   $s0 $a0
lw     $a0 24($fp)
bne    $a0 $zero label78
la     $a0 str_const33
```



```
li $t1 1
```

```
jal _dispatch_abort
```

There are four functions binary, decimal, hexa, hexadeci for conversion from Binary to decimal, Decimal to binary, Decimal to Hexa-decimal conversion, Hexa-Decimal to Decimal conversion respectively.

INCORRECT PROGRAMS:

1. Incorrect1.cl :

Code snippet which has error :

```
Id: String <- "Hello";
```

Violates the rule that object identifiers must start with a lowercase letter.(here Id violates)

Error log:

```
"incorrect1.cl", line 2: syntax error at or near TYPEID = Id
```

```
"incorrect1.cl", line 6: syntax error at or near TYPEID = Id
```

Compilation halted due to lex and parse errors

2. Incorrect2.cl :

Code snippet which has error :

```
id: String <- 'Hello';
```

```
str: String <- "Hello2
```

```
newline";
```

Violates the rule that non escaped newline characters should not appear and string should be in double quotes.

Error log:

"incorrect2.cl", line 2: syntax error at or near ERROR = '

"incorrect2.cl", line 4: syntax error at or near ERROR = Unterminated string constant

Compilation halted due to lex and parse errors

3. Incorrect3.cl :

Code snippet which has error :

```
- out_string(id);
```

Violates the rule that “-” is considered for single-line comments. Here have used just “-”.

Error log:

"incorrect3.cl", line 6: syntax error at or near '-'

Compilation halted due to lex and parse errors

4. Incorrect4.cl :

Code snippet which has error :

```
id1: Bool <- True;
```

Violates the rule that true and false are case-sensitive. But here we have used capital T.

Error log:

"incorrect4.cl", line 3: syntax error at or near TYPEID = True

Compilation halted due to lex and parse errors

5. Incorrect5.cl :

Code snippet which has error :

```
id: Str ing <- "Hello";
```

Violates the rule that there should not be blank space in the middle of keywords.

Error log:

"incorrect5.cl", line 2: syntax error at or near OBJECTID = ing

Compilation halted due to lex and parse error