GIT Department of Computer Engineering
CSE 222/505 - Spring 2015
Homework 04
Due to: 22.03.2016 11.59 pm

Write a program that converts expressions to assembly language.You have an input file that includes expression lines.So you will convert the file to an assembly file using java data structures.

Spesific Restrictions:

-Convert the infix expressions to postfix form.

-After getting postfix expressions, you should convert each operation to assembly instructions and registers.

-After all conversion you should save your assembly code to **.asm** file. You can test your code in mars editor.

-You should also check some unconditional situations (such as, undefined variable usage,division by zero, uncompitable type, out of register limit..).

-You should use only java list and stack and also you can create your data structures.

-You should use only these registers : [$t0-$t8,$a0,$v1] and instructions in table below .

Expression Spesifics:

-Expression variable type is integer.

-There are 5 operators(+,-,*,/,=) and print function.

You can directly use this assembly patches by tracking registers:

Expression	Assembly code
a b +	add $t3,$t2,$t1 #a in t1,b in t2
2 a +	li $t2,2 add $t3,$t2,$t1 #a in t1
2 a -	li $t1,2 sub $t3,$t2,$t1
b a *	mult $t1,$t2 mflo $t3 #get res from mflo
a 3 /	li $t3,3 div $t1,$t3 mfhi $t3
print c	move $a0, $t3 # print c,c in t3 li $v0, 1 #print_in syscall

available instruction table for homework

Name	**Usage**	**syntax**
Load immediate	li $t,c	$t = C (signed)
Move	move $t,$s	$t = $s
Add	add $d,$s,$t	$d = $s + $t
Subtract	sub $d,$s,$t	$d = $s - $t

Multiply	mult $s,$t	LO = (($s * $t) << 32) >> 32;
		HI = ($s * $t) >> 32;

Divide	div $s, $t	LO = $s / $t HI = $s % $t
Print	la $a0,$t	li $t3, 1 move $a0, $t3 # print t3. li $v0, 1 # syscall 1 = print_in syscall # do the syscall

OBJECTIVES:

- Preparing object oriented design for the problem
- Applying error handling
- Applying inheritance
- Applying code documentation
- Applying clean code standards
- Creating javadoc documentation

RESTRICTIONS:

- Use maven standard Project template
- Use only ArrayList data structure
- Can be only one main class in project
- Don't use any other third part library

GENERAL RULES:

- For any question firstly use course news forum in moodle, and then the contact TA.
- Use maven project management tool. And upload maven project into moodle.
- Code the Project in Java programming language. Java must be 1.8.* or bigger version.
- Any java IDE can be used in coding process.
- Implement all interfaces class
- Add all javadoc documentations for classes, methods, variables …etc. All explanation must be meaningful and understandable.
- Implement clean code standarts in your code;
    - Classes, methods and variables names must be meaningful and related with the functionality.
    - Your functions and classes must be simple, general, reusable and focus on one topic.
    - Use standart java code name conventions.
- Register github student pack and create private project and upload your projects into github.
- Your appeals are considered over your github project process.
- You can submitting assignment one day late and will be evaluated over forty percent (%40).
- Create report which include;
    - Your name, surname, studentid
    - Detailed system requirements
    - The Project usecase diagrams (extra points)
    - Class diagrams
    - Problem solutions approach
    - Test cases
    - Running command and results

GRADING :

- No OOP design	: -100
- No maven Project	: -100
- No error handling	: -95
- No javadoc documentation	: -95
- No clean code standard	: -95
- No report	: -90
- Disobey restrictions	: -98

- Your solution is evaluated over 100 as your performance. Don't forget this is performance project.