Project Description

Instructor: Dr. Michael Choi
TA: Chaoqi Ma

1. Project Overview

Create a numerical calculator. Refer to textbook chapter 2 and 3 for MIPS calculation and implementation.

2. Project Requirements

This project must be done under following environments:

Programming tool: **SPIM**

Virus Free Project

Under IIT department of Computer Science Policy, any project submission with virus should be excluded from the evaluation. It means, TA will not continue to evaluate your project material. The final grade of project will be 0. To avoid any potential debate, everyone should follow the department rule. Thus before you submit your project, you have to check your package with the latest antivirus software at multiple computers.

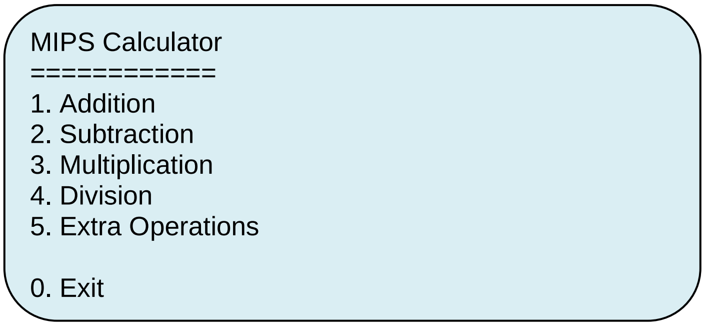
3. Description of the Project

User Interface: This project does not mandate a specific user interface. Thus you can freely design your own user interface. *Below screen shots are just samples and you don't need to follow the same way.*

Main Body: When your project starts, there are options for arithmetic operation such as Addition, Subtraction, Multiplication, and Division and operand numbers can be entered. Again, there is no interface requirement so you can implement your preferred way such as:

- human way such as "X + Y"
- Or Menu driven way:

- Example (no need to build the same menu interface – **Create your own interface**)



```
MIPS Calculator
============
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Extra Operations

0. Exit
```

Then, get inputs of operands such as X and Y.

Implementation requirements

Each arithmetic operation must be called out by a corresponding procedure from the main body. So MIPS assembler coding should include procedure calling parts and a remark or a comment must be included.

Main project requirements:

1. Get user selection menu which operation to perform basic arithmetic operation: Addition, Subtraction, Multiplication, or Division (Any interface format is fine but try human way ‘x + y’ or similar) and extras.

2. Calculator does not end unless the user selects Exit or required selection. Thus the program needs to go back to the beginning unless the user selects EXIT.

3. Reliable and robustness: if an incorrect input is entered such as “1 + &”, then a proper error message should be returned and keep continue to get next correct input instead of crash the calculator project.

4. Unless menu 0 is selected, the program does not end but keep working another operation by selecting menu.

Menu 5: Extra Operations

This selection provides extra features as you can find your scientific/engineering calculator. Those required features are:

```
MIPS Extra Features
================
1. Decimal to Binary          2. Decimal to Hexa
3. Hexa to Binary             4. Binary to Decimal
5. To Main Menu
0. Exit
```

1. You understand all other features.
2. Extra menu 1 to 4 returns conversion between those number system.
   e.g) decimal 5 returns to binary value of 101 by menu 1.
3. Selection 5 returns to the main menu: basic arithmetic operations
4. Selection 0 exits from the calculator (same from the main menu)

<u>What to submit?</u>

To the Course Blackboard project tab:

1. Source MIPS assembly codes (must be executable by SPIM)
   a. All source code must include detail comment of what the code does
   b. Detail comment is a good habit of documentation that professionals in the SW industry are required
   c. Good documentation means quality software in the industry
2. User’s manual document: How to use/run your project (include Program analyzing document) – because everyone’s interface is different so everyone needs to have a user documentation how to run your project
3. Final Design diagram document (or flow charts)
4. Self evaluation doc – describe what features are not yet implemented. If your project has all the features, no need to include it. But if your project does not include any feature, describe the item(s) in this document so your TA does not test it (them). TA will apply default penalty of each unimplemented feature/requirement. However, if your TA finds

any missing feature but not documented in your self evaluation, more penalty may be given.

All above must be in one zipped file (filename.zip).
** Only use zip format. **No other format such as RAR.**

Final Project Due Date: ***June 25, 2017 Sunday midnight*** or earlier
Submit to the class BB (blackboard)
Demonstrate your project to your TA on 6/26 or a separated appointment during next Lab session

- ~~Remote state and India campus people will be excluded in the demonstration due to physical participation difficulty~~
- ~~Submitting your project before the final due date is very important.~~

~~Another requirement to manage your project~~ (Waive for summer course)

~~As we know there are 9 steps of software development life cycle in CS401 or CS331 level. We need to practice it but this project simplifies those nine steps into two steps to encourage you to start your project as early as you can start and keep pace as you planned. Without having proper practice, you may have bigger problem in your professional career. As your professor and a mentor, I strongly recommend to follow another layer which you can manage your project well. ☺~~

~~While you are doing your project, there is an intermediate submission to encourage your progress. You should submit your current design document and your work schedule plan.~~

1. ~~**Design document**~~
~~Your design document shows your understanding of the project and diagrams of your coding. There is no formal design diagram but it needs to show your high and/or low design of your project. Your project schedule shows your progress.~~

2. ~~**Project Scheduling document**~~
~~You can divide project to your own small tasks such as high or low designing task, scheduling task, collecting requirements task, implementing task, testing and debugging task, and documenting task. You need to show how many hours (or days) are required to complete each task to complete your project. Per your schedule plan, you need to show by when you can complete a certain part of project or the whole project, by when you can complete testing, documenting, evaluating, debugging, etc.~~

~~25% of total project score will be earned by submitting the intermediate submission. Top qualified submission achieves 25%, middle quality documentations can achieve 20%, and poor quality submission achieves 15%. TA will compare all each document and compare to group top, middle, and poor quality groups or everyone may get full 25% if everyone's submissions are good enough.~~

~~Due date: **June 19, 2017 or earlier**, submit to the class BB~~