

CS2310 Computer Programming

Assignment Three (2020-21 Sem. A)

Deadline: **28-Nov-2020 23:59** (Week 13 Sat) No Late submission will be accepted

Problem description

In this assignment, you will create a simplified SpreadSheet program similar to Excel. The program will accept values or formula from keyboard and then it will print the numeric value in a cell upon query.

Similar to a real SpreadSheet program, the program behaves as follow:

- ☐ All cells are *Empty* by default
- ☐ Each cell could store an exact number or a formula referring to another cell
- ☐ Users could update the content of a cell later
- ☐ A cell's value will be updated when the cell that it refers to is updated
- ☐ The "value" of a cell which refers to an *Empty* cell is *Empty* as well
- ☐ Circular reference is considered as *Error* and is not allowed to happen

Unlike a real SpreadSheet program, the program is simplified as follows:

- ☐ The size of SpreadSheet is fixed at 26x26, with cells from A1 to Z26
- ☐ Non-*Empty* cells store only numeric value or formula (i.e. no string message)
- ☐ Formula only exist in the form of **<value> <op> <value>**, where:
 - **<value>** is either a number (e.g. 3.14) or another cell (e.g. A1)
 - **<op>** is +, -, * or /

You may use functions from **<string>** or **<cstring>** whenever appropriate.

Class construction:

To build the SpreadSheet program, you will construct a class called **Cell** which contains the following items:

(You are not allowed to change the specifications below, but you can add in new members or functions if needed).

Private attributes / methods:	
double v1, v2;	Immediate numeric values contained in the formula. E.g. , if the formula is 3 + 5 , then v1=3, v2=5 . The values should be 0 in empty cells, or when the corresponding position is not used. E.g. if the formula is A9 * 7 , then v1=0, v2=7 . When the cell is a single value (e.g. 3.14), The value is stored in v1 . v2 stores 0.

char op;	Operator in the formula (+, -, *, /). If the cell is Empty, or when the cell is a single number, op will be NULL (i.e. ASCII 0)
Cell *c1, *c2;	Pointers (physical address) to the cells being referenced in the formula. The pointers are NULL for Empty cells, or when the position is not a reference. E.g., if the formula is 3 * A1 , then c1 is NULL and c2 points to the address of Sheet[0][0] . (where Cell Sheet[26][26]; is the array you create in main())
bool empty	Whether the cell is <i>Empty</i> (having no number/formula)

Public attributes / methods:	
Cell()	Default constructor setting all members to zero / NULL
bool isEmpty()	Report whether the Cell is Empty, or directly/indirectly refers to another Empty cell
double getValue()	Should be called only when isEmpty() is false. It returns the final numeric value of the cell. Eg. If A1 is 3 , and A2 is 2 * A1 , then calling the getValue() of Cell A2 will return 6
bool refers(Cell *c)	Whether the cells directly/indirectly refers to another Cell c Eg. If A1 is 3 , and A2 is 2 * A1 . When you pass A1 's address (pointer) to A2 's refers() function, it will give true
bool setFormula(char* F, Cell Sheet[26][26])	<p>Update the member variables / pointer using the cstring F. Cell Sheet[26][26] is the array you created in main(). It is passed to the function to facilitate getting the pointer of the referenced cells. (e.g. <i>&Sheet[0][0]</i> is the address of A1). The Sheet[][] array should be accessed in a read-only manner.</p> <p>You may assume cstring F is always syntactically valid (e.g. "3", "B7", "A2 + 4", "A3 - B7"...etc) However, when the formula generates circular reference, the setFormula() should DO NOTHING and return false. For example, if A1 is currently A2 + 1. When A2's setFormula() is called with "A1 + 1", then you should discard the update and return false.</p> <p>Hint: For all C++ objects, there is a default variable this, which is the pointer (Cell *), pointing to the object whose function is currently being called.</p>

Input / Output Specification:

Your program should keep running, until the “END” command is entered. Each line contains exactly one command and it’s guaranteed that at least one space between cells / numbers / punctuations. Occasionally there might also be extra space(s) at the end of a line, *however, there is no space within a cell address, for instance, **A15** is never written as “A 15”.*

Syntax	Meaning
<cell> = <value/cell> e.g. A1 = 5	Set the value of cell A1 to 5 , overwriting old value / formula if any. No output is needed.
<cell> = <value/cell> <op> <value/cell> e.g. A1 = 3 * A2	Update the formula of the cell A1 to 3 * A2 . Usually no output is need. However if that formula generates a circular reference, the program should print “ERROR: Circular reference!” You may ignore the “division by zero” case and assume that it will never happen.
<cell> ? e.g. A1 ?	Print the numerical value of the cell A1. You just need to use the default setting of cout for printing. There’s no need to care about the number of significant digits or scientific notation. Print “is Empty” if the cell is empty or directly / indirectly refers to an empty cell.
END	Terminate the program.

Sample Input / Output

Example 1: (User Input is underlined)	Example 2: (User Input is underlined)
<u>A1 ?</u> A1 is Empty <u>A1 = 5</u> <u>A1 ?</u> A1 is 5 <u>A2 = A1 + A1</u> <u>A2 ?</u> A2 is 10 <u>END</u>	<u>B5 = B4</u> <u>B5 ?</u> B5 is Empty <u>B4 = 1 / 5</u> <u>B5 ?</u> B5 is 0.2 <u>END</u>

Example 3: (User Input is underlined)	Example 4: (User Input is underlined)
<u>A1 = 5</u> <u>A2 = A1 + A1</u> <u>A3 = A2 + A2</u> <u>A4 = A3 / 7</u> <u>A3 ?</u> A3 is 20 <u>A4 ?</u> A4 is 2.85714 <u>END</u>	<u>A1 = 3</u> <u>A2 = A1</u> <u>A3 = A2</u> <u>A3 ?</u> A3 is 3 <u>A1 = A3</u> A1 ERROR: Circular reference! <u>A1 ?</u> A1 is 3 <u>A1 = 7</u> <u>A3 ?</u> A3 is 7 <u>END</u>

Marking criteria

Submitted program will be tested repeatedly with PASS. Marks will be graded objectively based on the number of correct outputs reported by PASS.

- If the program is not compilable, zero mark will be given.
- Make sure that the output format (spelling, spacing...etc) follows *exactly* the sample output above otherwise PASS will consider your answer incorrect. Note that the marker will make NO manual attempt to check or re-mark program output.

Unlike tutorial exercises, for assignment, PASS will NOT make ALL test cases visible online. (*i.e. there are hidden test cases*). You should not assume that the report generated from the "Test" button is your final result.


Note that the marking in PASS is automatic, therefore the following situations may lead to extremely low or zero marks:

- Submission of non **.cpp** files (*e.g. .exe or .zip files*)
- Submission with the "**test**" function rather than the "**submit**" function
- Input/output does not match the requirements as defined. (*e.g. incorrect spacing, incorrect spelling, letter cases or punctuations*)

Please also note that the PASS plagiarism check will also be turned on during assignment marking. The same penalty will be applied without distinguishing which one is the source / copier. Please safeguard your files if you work on your assignment using public computers (e.g. CityU Lab)

4. Testing and Submission

Students should submit **separate** cpp files for **part a)** and **part b)** to PASS before the deadline. **No other submission method (e.g. hardcopy, email...etc) is accepted.** Students may use whatever IDE/compiler for development, but programming using non-standard, platform specific features (which is not compilable in MS Visual Studio) will lead to zero mark. The marker will make no attempt to fix the syntax error and make the code compilable.

It is advised that students test the programs with PASS (*using the*  *function*) before final submission (*with the “submit” button*). Be warned that the system could be extremely busy and may become sluggish in responding near deadline. Only **.cpp** files are accepted. Do not submit **.obj** , **.exe** or any other files.

To protect yourself, it is advised that you write down your particulars (full name, student number, eid...) in the beginning of your source code *as comment*.