

**EE308FZ Software Engineering**

**LAB 5 On-site programming of team projects**

**Task**

The on-site programming requires the development of a oral calculation program, which covers students of all ages in primary schools and improves the quick calculation ability of pupils' addition and subtraction (multiplication and division).

**Functional requirement**

1. Applicable age range of oral calculation software: primary school students of all ages (primary school grade 1 to primary school grade 6);
2. Different difficulty levels are set according to students of different ages. The difficulty levels are as follows:
   1. Grade 1 to grade 2: addition and subtraction of integers within 100;
   2. Grades 3 to 4: addition, subtraction, multiplication and division of integers;
   3. Grades 5 to 6: addition, subtraction, multiplication and division including decimals.
3. The test questions are automatically and randomly generated by the software;
4. The number of test questions is determined by the user;
5. After the answer is finished, the software automatically judges the user's scores.

**Reference sample**

Sample 1 :

Output: Please enter your grade?

Input: grade 2

Output: Please enter the number of questions?

Input: 2

Output: 10+24=?

Input: 34

Output: 33+44=?

Input: 87

Output: end! Wrong question, your score is 50.

Sample 2 :

Output: Please enter your grade?

Input: grade 6

Output: Please enter the number of questions?

Input: 3

Output: 10.2+21.55=?

Input: 31.75

Output: 12.4 \* 22=?

Input: 272.8

Output: 150/3=?

Input: 50

Output: end! All right, that's great! Your score is 100.

Note: The words "Input:" and "Output:" in the sample do not need to be displayed in the specific coding.

**Implementation requirements**

1. The team needs to learn and use Github to collaborate, and everyone needs to have their own submission record. Note: Github Learning Link:
   * <https://www.cnblogs.com/schaepher/p/5561193.html>
   * <https://www.cnblogs.com/schaepher/p/4933873.html>
2. The software can be presented by command line or interface display.

**Blog requirements (published as a team)**

1. Basic format [see lab1];
2. Give the division of responsibilities of team members;
3. Give the program running environment;
4. Give a screenshot of the software running;
5. Give at least 3 events that take a long time in coding, arguing, reviewing and other activities and give you great gains;
6. Team programming experience;
7. PSP and Learning Progress Bar Reference: <http://www.cnblogs.com/vertextao/p/7469789.html>;
8. Give GitHub link and commit record.

**Grading standards（100）**

Teaching assistant rating:

1. Code function completion (50%)

2. Github collaboration data and indicators (30%)

3. Presentation effect and blog description (20%)

The team leader should grade each student's contribution. The rule is that the total score of the whole group = scores of each group \* number of people. And the scores of each student in the group cannot be the same.

**Time management**

| Time | Arrange |
| --- | --- |
| 10/18 15:50 | Release the content of on-site programming, and each team  can prepare in advance. |
| 10/18 15:55 | Start programming. |
| 10/18 18:30 | TA tests software functions and scores |

**Tips:**

1. TA can’t notice all students. If your blog can impress TA, you can get higher remark.

2. If you have some question, you should ask directly in the QQ group and you’d better not make a private chat with TAs or tutor.

3. Please issue your homework in the section of assignment on CSDN.

**4. The Link of each assignment you would accomplish must be submit to Moodle, or you will lose the score of the assignment.**

**5. Each assignment submitted to CSDN must be completed in English.**