C++ 基础与深度解析
第三章作业思路提示
2022-11-16

主讲人 天哲

纲要

深蓝学院
shenlanxueyuan.com

- 作业提交
- 程序重点
- 常见问题
- 批改标准

作业提交

深蓝学院 shenlanxueyuan.com

- 「提交」文件命名为：用户名-第n章作业，回答框简要介绍作业即可，作业内容放在附件中上传。
- 「格式」附件格式可以为pdf或zip压缩包，建议word等格式的文档导出为pdf再提交。
- 源代码后缀名使用.cpp/.cxx/.cc/.c++ 把源文件放到附件中，不要把源文件粘贴到作业说明中！！！
- 提交前自己稍微测试一下，有收到不能编译或者错误很明显的代码....
- 每个助教要批改上百份代码，尽量不要让他们帮你修复编译错误

正确示例

张三-第3章作业.zip/ZhangSan-Proj3.zip

|- source.cpp
|- header.h
|- readme.txt/document.doc/readme.pdf ...
|- screenshots.jpg/png....
|- 其他文件.....

错误示例

homework.rar

源代码.pdf

source.txt

程序重点 -- 数据结构

等式的表达，可以分开，也可以使用结构体

```
// num1 (op) num2 = result;
int num1 = 0, num2 = 0, result = 0;
char op = '+';
const char operations[] = {'+', '-', '*', '/'};

// 也可以使用结构体
// operation 0,1,2,3 represents for +,-,*,/
struct Question
{
    int num1;
    int num2;
    int operation;
    int result;
    //int userinput 记录用户的答案
    //void print () //输出题目
};
```

```
const int maxNumber = 100;
//使用数组的话，可以提前开辟一个比较大的空间
int array_num1[maxNumber], array_num2[maxNumber], array_result[maxNumber];
Question array_questions[maxNumber];

//使用vector的话，后面还可以调整容量，比较灵活
std::vector<int> vec_num1, vec_num2, vec_result;
std::vector<Question> vec_questions;
```

程序重点 -- 随机数产生 和 计时

深蓝学院 shenlanxueyuan.com

```
//随机数的产生
/* initialize random seed: */
srand(time(0));
int range = 100;
int ranNum = rand() % range; //产生0-99的随机数

//计时的实现
const std::chrono::time_point<std::chrono::steady_clock> start =
    std::chrono::steady_clock::now();
// do something....
//使用auto 来避免繁复的输入
const auto end = std::chrono::steady_clock::now();
auto duration = std::chrono::duration_cast<std::chrono::seconds>(end - start).count();
```

对于clock() 有时会失效
https://stackoverflow.com/questions/55322820/how-can-i-measure-time-with-clock-when-i-use-cin
https://stackoverflow.com/questions/36091472/clock-doesnt-work-correctly

常见问题 --非法输入的检查

如果不检查非法输入，可能会影响正常答题，甚至程序崩溃

```
int read_int ()
{
    int input;
    std::cin>>input;
     while(std::cin.fail())
    {
        std::cin.clear();
        std::cin.ignore();
        std::cout << "输入格式错误，请重新输入\n";
        std::cin >>input;
    }
    return input;
}
```

```
int input_int() //判定是否是整数
{
    std::string in_answer;
    std::cin >> in_answer;
    while (in_answer.find_first_of("0123456789") == std::string::npos
           || in_answer.find_first_not_of("0123456789") != std::string::npos)
    {
        std::cout << "输入格式不对，请确认后再次输入\n";
        std::cin >> in_answer;
    }
    return stoi(in_answer);
}
```

常见问题 -- 除法


```
num1 = rand() % range;
num2 = rand() % range;      // 1.会有除数为0的问题
num2 = rand() % range + 1; //解决问题1
result = num1 / num2;       // 2.会有35/3=10的问题，
//3.同时大约有50%几率 num2>num1 会使答案为0
```

```
//方案1，如果不满足条件，不停循环尝试
    while (num2 == 0 || num1 % num2 != 0)
    {
        num1 = rand() % range;
        num2 = rand() % range;
        result = num1 / num2;
    }
```

```
//方案2，使用一个乘法反向推算
    result = rand() % range;
    num2 = rand() % range + 1;
    num1 = num2 * result;
```

....其实方案2也并不是完美，例如难度会从0/1到9900/100这样难度跨度太大了，但相比初始方案已经好太多了

具体实现中 只要自圆其说都可以。例如有个同学在除法中 答案要求输入 商+余数的形式 我觉得也很不错

进阶问题 – 一些建议

- 不要复制粘贴多行代码，如果你发现需要复制粘贴，考虑抽象成一个函数
- 复杂的函数给一些注释
- 使用标准库，不用使用<windows.h>
- 使用utf-8格式保存源码

```
//This function  print one question
//if result is not null, it prints also the result
void print_one(const int num1, const int num2, const int operation, const int * const result = nullptr)
{
    //operations
    const char operationschar[] = {'+', '-', '*', '/'};
    //show question
    std::cout << num1 << ' ' << operationschar[operation] << ' ' << num2 << std::endl;
    if (result)
        std::cout << "Correct Answer is " << *result << std::endl;
    return;
}
```

批改标准

- 完成基本功能 → 合格
- 完成部分扩展，或完成全部扩展，但是有明显的BUG → 良好
- 完成全部扩展，并且没有明显的BUG （例如在本章中，要求输入检查和防止被除数为0）→ 优秀
- 如果对批改结果有异议，可以使用“追问”系统

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening