

程序设计基础：第一次实验题

开始时间：2023.10.22 22:00

截止时间：2023.11.05 23:59

引言

欢迎各位同学来到程序设计基础课程的实验题部分！

在每次实验题的练习中，各位同学将在文档的指引下，通过两个模块深入探索 C++ 中的语法细节，并利用获得的知识开展一个有趣的小编程，感受如何用程序语言搭建出一个实用的系统。我们期待大家能够立足课上的知识点，更加了解程序语言的设计思想，并锻炼资料查找和实际编程的能力。实验题中的知识对于大作业的完成也会起到帮助！

所有需要回答的问题或完成的工作均在文档中通过黑色的点标记了出来。在完成作业后，你需要提交一个报告来总结自己的发现和编程结果。每次实验的满分为 10 分，其中两个模块均占 4 分，最后的小编程占 2 分。在评分中，我们会考察细节探索的全面性、系统编写的完整性和语言表述的简洁性。因此，同学们只需积极完成作业即可，无需通过更长的报告篇幅或超出要求的扩展功能来体现你的工作量。

作业文档的解释权归属于教学团队，如果你对文档有疑问，或者找到了文档中的错误，请与助教联系！祝愿大家在实验题中玩得开心，有所收获！

模块一：随机数的产生

使用课上学习的逻辑思维，我们能够描述很多确定性的过程。然而，生活中还有大量的过程是有随机性的，例如抽奖的结果、游戏中的暴击、信息的加密等。在后续的数学课和专业课中，大家会学习如何对它们进行建模，而在本门课中我们关注的问题是，应该如何来搭建有随机性的程序？

产生随机性的最重要的基础便是随机数的生成。在 C/C++ 语言中，`rand()` 函数能够产生随机数，其在 C 语言中位于 `stdlib.h` 头文件中，在 C++ 语言中则位于 `cstdlib` 头文件中。因此，我们可以很轻松写出一个产生随机数的程序：

```
1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4
5 int main() {
6     cout << rand() << endl;
7     return 0;
8 }
```

很好，程序输出了一个数字！但事情好像并没有那么简单，很多问题浮现了出来，请你通过编写代码或者查找资料尝试回答：

- **rand()**函数是通过什么机制生成的随机数？在你的机器中，其中的参数是多少？这种机制的优势是什么？
- 生成的数字真的是随机的吗？通过代码验证你的想法。
- 生成的随机数范围是多少？如何生成一个特定范围内的随机数？

在查找资料的过程中，你或许发现了一个和随机数函数长得很类似的**srand()**函数。它与**随机种子**这个概念息息相关。事实上，不只是 C++，在当下很火热的深度学习框架中，这也是一个很重要的概念。请你尝试回答一下：

- 什么是随机种子？在 C++ 程序中要如何设定？
- 设定随机种子后有什么效果？通过代码展示一下。

现在你应当理解了 C++ 中最常见的产生随机数的方法和原理了。不过由于其仍有一定的缺陷，所以也有很多其他产生随机数的方法（比如 C++11 标准中的**<random>**头文件）。

- C++ 中还有哪些产生随机数的方法？尝试简单概述一下。

模块二：多样化程序输出

这个模块中我们来讨论学习程序设计时第一个接触的知识点：**输出**！

我们此前学过，C++ 中的输出语句为：

```
1 cout << "Hello world!" << endl;
```

而如果你喜欢 C 语言风格的输出，则是：

```
1 printf("Hello world!\n");
```

第一次接触时，你可能是通过死记硬背记住了这个语法，今天让我们探索一些其中的细节。请你通过编程尝试或者查阅相关资料，回答以下问题：

- 除了 cout 以外，使用 C++ 的流式输出还能输出到哪里？为什么要区分这几个？它们的区别和特点是什么？
- 如何在 cout 和 printf 中设置特定的输出格式？（例如输出特定的数字位数、对整数进行填充、科学计数法、不同进制的输出、控制对齐方式等）

此外，命令行黑白的输出未免有些单调了。在 ANSI 标准中，以**ESC**（ASCII 码 27，转义字符 \e）开始的字符串可以表达特殊的输出格式，即 ANSI 转义序列（ANSI escape sequences）。其格式形如**\033[n1;n2;n3mXXX\033[0m**，其中 n1、n2、n3 为不同的样式代码，且可以包含任意项（即可以不止 3 项），XXX 为文本内容。

- 如何在命令行中输出彩色的文字（即设置输出文本的前景色和背景色）？请用代码展示一下你的发现。

Note: 如果你使用的是早期的 Windows 版本，可能看不到彩色的效果，可以尝试在终端中输入如下指令修改注册表，打开 ANSI 颜色输出的支持。

```
1 reg add HKEY_CURRENT_USER\Console /v VirtualTerminalLevel /t REG_DWORD /d 0x00000001 /f
```

现在，你应该可以自如地进行程序的输出了。这其中还有一些更深的问题，以当前的知识还不足以解释。比如 `endl` 是什么？输出它的时候发生了什么？为什么 `<<` 可以连续地输出字符串、整数、浮点数等各种形式，而 `printf` 就必须指定格式？本次实验中你无需解答，但如果你未来选修《面向对象的程序设计基础》，学过重载运算符和函数模板等概念后，不妨再来试试找寻它们的答案。

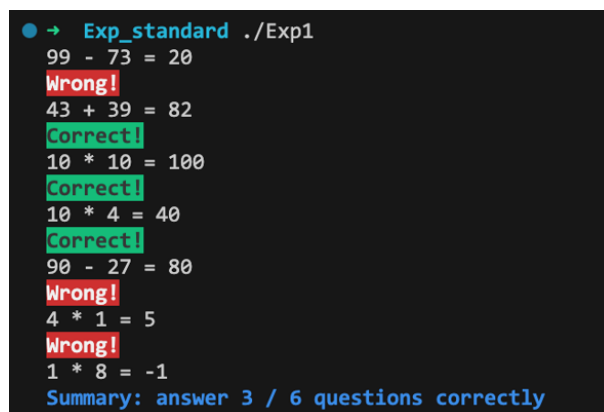
程序的正确性和稳定性是基础，而其与用户的交互也同样十分重要。希望这个模块的探索能让你写出更加易于交互的程序！

小程序：一个口算训练程序

小学学习数学时，口算是一个重要的训练方法。然而出大量的口算题不但枯燥，批改它们同样要耗费不少精力。一名小学生的父母向你发出了委托，希望你编写一个程序帮助他们的孩子训练口算。他们的需求包括：

1. 进入程序后不断生成口算题目。每次随机生成一个 100 以内的加减法或 10 以内的乘法表达式。表达式的答案必须是正整数，且不超过 100。
 2. 要求用户输入答案，程序需要对答案的正确性进行判断。
 3. 判断后需要告诉用户答案的正确性。如果回答正确则显示绿色底色的提示语，如果回答错误则显示红色底色的提示语。
 4. 用户可以输入-1 结束训练。结束训练后，输出答题情况的总结，总结需要是蓝色的字。
- 请你利用之前两个模块的知识，编写代码实现这个程序。请在报告中概述你的代码逻辑，并用录屏展示系统运行的效果（详见[作业提交要求](#)章节）。

Note: 一个样例如下图所示。（你的程序满足以上需求即可，不需要和这个样例完全一致）



```
● → Exp_standard ./Exp1
99 - 73 = 20
Wrong!
43 + 39 = 82
Correct!
10 * 10 = 100
Correct!
10 * 4 = 40
Correct!
90 - 27 = 80
Wrong!
4 * 1 = 5
Wrong!
1 * 8 = -1
Summary: answer 3 / 6 questions correctly
```

一些建议

- 互联网中能够获得丰富的知识。C++ 参考手册网站上能查到很多函数或头文件的含义和用法，一个常见的网站是en.cppreference.com。使用维基百科（en.wikipedia.org）等相对权威的资源也是一个不错的选择。此外，知乎、CSDN、StackOverflow 等博客或问答网站通常也有着很详细的教程和指导。

- 我们欢迎同学之间关于实验题目的交流，这是学习知识的重要手段。但请注意不要违反学术规范，即不要直接复制粘贴他人实验报告的答案！请确保报告中的每一句话和每一行代码都是你自己的劳动成果！

作业提交要求

你需要提交两个文件：一个**PDF 文档**，包含两个模块的所有书面题目结果和小编程的代码逻辑思路；一个**录屏文件**（不超过 2 分钟），展示小编程的系统运行效果。请将二者打包在一个**ZIP 压缩包**中，命名为**【学号-姓名-lab1.zip】**，并在截止日期之前通过**网络学堂**提交。

我们为实验留出了足够长的时间，所以请及时完成作业并**遵守作业的截止时间**。教学团队不受理截止时间后的补交。如果你遇到了极其特殊的情况，客观上无法完成实验（如生病、因公出差、参与重要活动等），请通过网络学堂答疑区或发送邮件至wt22@mails.tsinghua.edu.cn向助教说明，我们会根据具体情况与你讨论实验的提交要求。

写在最后

本次实验是教学团队首次命题，我们希望尽可能做到知识性和趣味性的平衡，让大家收获更丰富的知识和探索的能力。但初次尝试难免可能有不详尽之处，如果同学们对实验的安排或内容有想法或建议，欢迎向助教团队反馈，帮助我们将课程建设地越来越好！