计算机高级语言程序设计实习题目

实习一、简单程序设计

实习要求

1. 实验内容
   开展C++简单程序设计，掌握简单程序设计的方法（自顶向下）、输入输出流的使用、变量的定义与使用以及程序的基本结构等内容的实践。
2. 实验要求：
   - 每一题单独建立一个文件夹，如：1_1,1_2...，只包括源代码(.h,.cpp)文件和工程文件（若有）
   - 上传不大于10M的报告文件(*.doc, *.docx)

题目1. 输出菱形

输入菱形的边长（下图菱形边长为10），在屏幕输出如下图所示的菱形。

```
          *
         ***
        *****
       *******
      *********
     ***********
    *************
   ***************
  *****************
*******************
  *****************
   ***************
    *************
     ***********
      *********
       *******
        *****
         ***
          *
```

题目2. 简单运算器

输入两个整数与运算符，输出运算结果。如：

- 输入a+b，程序输出运算结果
- 输入a-b，程序输出运算结果
- 输入a*b，程序输出运算结果
- 输入a/b，程序输出运算结
- 将程序改造为循环计算，直到输入a#b结束程序

题目3. 输出九九乘法表

输出九九乘法表，要求成下三角，并且=与结果右对齐

示例如下：

```
1*1=1
2*1=2   2*2= 4
3*1=3   3*2= 6  3*3= 9
4*1=4   4*2= 8  4*3=12  4*4=16
5*1=5   5*2=10  5*3=15  5*4=20
```

题目4. 猜数游戏

程序随机生成一个99以内的非负整数，提示用户所输入的数 与 随机生成的数的大小关系。用户根据提示，直到输入的数与随机数相等时退出猜数游戏。（循环结束条件可以限定输入的最多次数，达到最大次数未拆对，则游戏失败。）

> 随机数生成代码如下：

```cpp
#include <iostream>
#include <ctime>        //time
#include <cstdlib>      // rand
using namespace std;

int main()
{
    srand((unsigned int)time(0));     //随机数种子
    cout << rand()%100 << endl;       //0-99间的随机数
    return 0;
}
```

实习二、面向过程程序设计

题目1. 日历输出

输入年、月，输出如下所示的日历。在设计程序时，分析输出日历涉及几个函数，在Calendar.h声明函数，在Calendar.cpp中实现函数，在main.cpp中调用日历输出函数。

```
          May 2022
------------------------------
| Sum Mon Tue Wed Thr Fri Sat |
|               1   2   3   4  |
|  5   6   7   8   9  10  11  |
| 12  13  14  15  16  17  18  |
| 19  20  21  22  23  24  25  |
| 26  27  28  29  30          |
------------------------------
```

题目2. 高阶计算器

题目：编写一个整数计算器程序，支持：+、-、*、/、x^y(x的y次方)、！(阶乘)的混合运算。

功能：

- 判断算式是否合法(操作符号和操作数的数量、位置)
- 支持混合算式（输入的算式并不预先设定有多长，通过字符流来解析），如：1+2+5*3
- 运算表达式可不考虑（）的情况，但运算的优先级需要考虑
- 用户输入#，程序退出

题目3. 简易成绩管理系统

模拟成绩信息管理系统，在控制台模拟成绩信息的菜单，完成相应的指令：

```
    A - Add a score 添加一个成绩
    D - Del a score 删除一个成绩
    L - List All scores and count 列出所有成绩
    M - Minimum of scores 成绩最小值及索引
    N - Average of scores  成绩均值
    P - Maximum of scores 成绩最大值及索引
    T - Standard derivation of scores 成绩标准方差
    S - Save scores to file 保存成绩到文件 (选做)
    O - Open scores from file 从文件加载成绩（选做）
    X - Exit 退出程序
```

> 提示：成绩用数组存放，每个功能用函数实现。

实习三、面向对象程序设计

题目1. 分数类fraction

实现分数类fraction的功能，完成简单的运算。

数据成员包括：分子和分母（整数）

成员函数包括：构造函数、分数简化、分数与double的强制转换，输入输出操作符重载，分数运算（包括分数与分数之间的+-*/运算，分数与整数的+-*/运算，浮点与分数的+-*/运算）

最关键的是，以下的测试代码能正确运行。

实例代码

```
#include <iostream>
#include <fraction.h>   // class fraction

using namespace std;

int main()
{
    fraction    f1(1, 2);   // 表示1/2
    fraction    f2(1.2, 0.5);   // 表示1.2/0.5
    fraction    f3(0.6);        //转换为分数3/5
    fraction    f4;
    fraction    f5 = f3;        //拷贝构造

    cin >> f4;  //输入分子，分母，并简化

    cout << f1 + f2 << endl;
    cout << f1 - f2 << endl;
    cout << f1 * f2 << endl;
    cout << f1 / f2 << endl;

    cout << f3 + 0.5 << endl;   // -*/都支持
    cout << 0.5 + f3 << endl;   // -*/都支持

    cout << f4 + 1 << endl;
    cout << 1 + f4 << endl;

    cout << double(f5) + 0.5 << endl;

    return 0;
}
```

要求

1. 完成类fraction 的声明，编写文件 fraction.h
2. 实现类的功能，编写文件 fraction.cpp
3. 代码符合编码规范，有清楚的注释
4. 测试所有的函数，确保无误

题目2. 魔兽格斗

怪物格斗游戏的目标是在玩家死亡或升到20级之前尽可能多地收集金币。

游戏规则

怪物格斗游戏的规则如下：

- 玩家每次会遇到一个随机生成的怪物。
- 对于每个怪物，玩家有两个选择：（R）un或（F）ight。
- 如果玩家决定逃跑，他们有50%的机会逃脱。
- 如果玩家逃脱了，他们就会进入下一次遭遇战，不会有任何不良影响。
- 如果玩家没有逃跑，怪物会得到一次自由攻击，玩家可以选择他们的下一个行动。
- 如果玩家选择了战斗，玩家就会先攻击。怪物的生命值会被玩家的伤害减少。
- 如果怪物死了，玩家会拿走怪物身上的所有金币。玩家也会提升等级，使其等级和伤害增加1。
- 如果怪物没有死，怪物会反过来攻击玩家。玩家的健康状况会因怪物的伤害而减少。
- 当玩家死亡（损失）或达到20级（胜利）时，游戏结束。
- 如果玩家死亡，游戏应该告诉玩家他们的等级和他们有多少金币。
- 如果玩家赢了，游戏应该告诉玩家他们赢了，以及他们有多少金币。

游戏环节的示例

下面是一个游戏环节的示例：

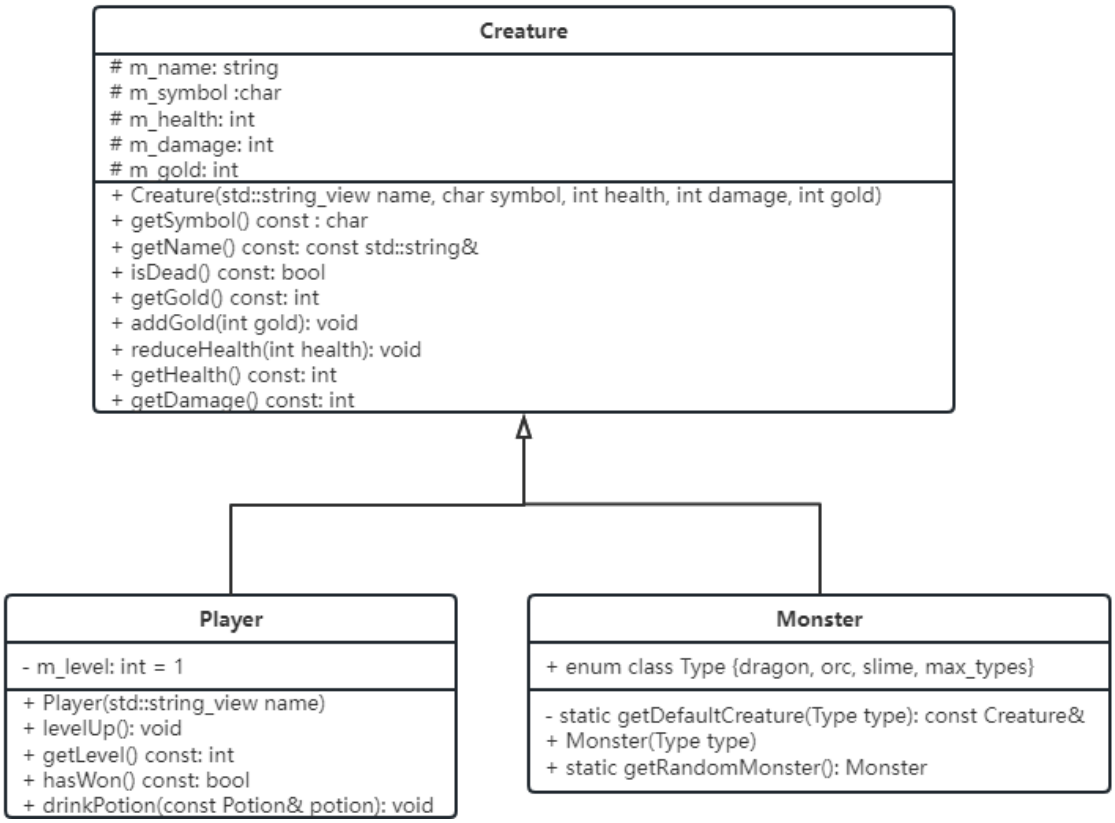
> Enter your name: Alex
> Welcome, Alex
> You have encountered a slime (S).
> (R)un or (F)ight: f
> You hit the slime for 1 damage.
> You killed the slime.
> You are now level 2.
> You found 10 gold.
> You have encountered a dragon (D).
> (R)un or (F)ight: r
> You failed to flee.
> The dragon hit you for 4 damage.
> (R)un or (F)ight: r
> You successfully fled.
> You have encountered a orc (o).
> (R)un or (F)ight: f
> You hit the orc for 2 damage.
> The orc hit you for 2 damage.
> (R)un or (F)ight: f
> You hit the orc for 2 damage.
> You killed the orc.
> You are now level 3.
> You found 25 gold.
> You have encountered a dragon (D).
> (R)un or (F)ight: r
> You failed to flee.
> The dragon hit you for 4 damage.
> You died at level 3 and with 35 gold.
> Too bad you can’t take it with you!

- 提示：创建4个函数

- main()函数应该处理游戏设置（创建玩家）和游戏主循环。
- fightMonster()处理玩家和单个怪物之间的战斗，包括询问玩家他们想做什么，处理Run或Fight的情况。
- attackMonster()处理玩家对怪物的攻击，包括提升等级。
- attackPlayer()处理怪物对玩家的攻击。

类设计

程序将由3个类组成：Creature（生物）、Player（玩家）和Monster（怪物）。Player和Monster都继承自Creature。类的关系和内容如下图所示。



class Creature

- Creature有5个私有属性：
  - 一个名字name（std::string），
  - 一个标志symbol（char）
  - 一个健康量health（int）
  - 每次攻击的伤害量damage（int）
  - 携带的金币量gold（int）
- 编写一套完整的属性获取函数（每个成员都有一个get函数）。
- 添加另外三个函数：
  - void reduceHealth(int) 将生物的健康状况减少一个整数。
  - bool isDead() 当生物的健康状况为0或更少时返回true。
  - void addGold(int) 向生物添加黄金。

class Player

- Player类
  - Player类继承于Creature
  - Player有一个额外的成员，即玩家的等级level，从1开始
  - Player有一个自定义的名字（由用户输入），symbol初始化值'@'
  - Player有10个健康，开始时有1个伤害，没有黄金

- 函数成员
  - 编写一个名为levelUp()的函数，使玩家的等级和伤害增加1
  - 为level成员编写一个get函数getLevel()。
  - 编写一个名为hasWon()的函数，如果玩家达到20级，则返回真。
- 写一个新的main()函数，询问用户的名字并产生如下输出：

> Enter your name: Alex
> Welcome, Alex.
> You have 10 health and are carrying 0 gold.

class Monster

- 在Monster类中添加一个名为Type的枚举，该枚举包含了这个游戏中要用到的3种怪物的枚举器：dragon、orc和slime（你还需要一个max_types的枚举器，因为这在后面会很有用）。

```
class Monster : public Creature
{
public:
    enum class Type
    {
        dragon,
        orc,
        slime,
        max_types
    };
};
```

- 每种怪物类型将有不同的名称、符号、起始健康状况、金币和伤害。下面是每个怪物类型的统计表：

Type	Name	Symbol	Health	Damage	Gold
dragon	dragon	D	20	4	100
orc	orc	o	4	2	25
slime	slime	s	1	1	10

编写一个Monster构造函数，这样我们就可以创建Monster。Monster构造函数应该接受一个类型枚举作为参数，然后为该种怪物创建一个具有适当成员变量的Monster对象。

使用一个查找表。一个查找表是一个保存所有预定义属性的数组。我们可以根据需要使用查找表来查找某个特定怪物的属性。那么，我们如何实现这个查找表呢？这并不难。我们只需要一个数组，为每个怪物类型包含一个元素。每个数组元素将包含一个Creature，其中包含该类型怪物的所有预定义属性值。我们把这个数组放在Monster的一个静态成员函数里面，这样我们就可以为一个给定的Monster::Type获得一个默认的Creature。

- 查询表的定义如下：

```cpp
// As a private member of Monster
static const Creature& getDefaultCreature(Type type)
{
  static const std::array<Creature, static_cast<std::size_t>
(Type::max_types)> monsterData{
    { { "dragon", 'D', 20, 4, 100 },
      { "orc", 'o', 4, 2, 25 },
      { "slime", 's', 1, 1, 10 } }
  };

  return monsterData.at(static_cast<std::size_t>(type));
}
```

- 为Monster添加一个名为getRandomMonster()的静态函数。

  这个函数应该从0到max_types-1中选取一个随机数，并返回一个具有该类型的怪物（通过值）（你需要将int静态化为一个Monster类型，并将其传递给怪物构造函数）。

挑战Potion(选做)

假如玩家没有磨好他的剑，不足以打败强大的龙。通过实施以下不同尺寸的药水来帮助他：

Type	Effect (Small)	Effect (Medium)	Effect (Large)
Health	+2 Health	+2 Health	+5 Health
Strength	+1 Damage	+1 Damage	+1 Damage
Poison	-1 Health	-1 Health	-1 Health

可以自由发挥创意，添加更多的药水或改变它们的效果！

玩家有30%的机会在每次战斗胜利后找到一种药水，并可以选择喝或不喝。如果玩家不喝药水，它就会消失。玩家不知道找到的是什么类型的药水，直到玩家喝下它，这时药水的类型和大小就会显示出来，效果也会被应用。

在下面的例子中，玩家发现了一个毒药，并因喝下它而死亡（在这个例子中，毒药的伤害性更大）。

> You have encountered a slime (s).
> (R)un or (F)ight: f
> You hit the slime for 1 damage.
> You killed the slime.
> You are now level 2.
> You found 10 gold.
> You found a mythical potion! Do you want to drink it? [y/n]: y
> You drank a Medium potion of Poison
> You died at level 2 and with 10 gold.
> Too bad you can't take it with you!

- 提示：添加一个药水类，它有一个type和size的成员变量，以及一个返回其名称的成员函数和一个创建随机药水的静态成员函数，类似于getRandomMonster()函数。
  在Player类中，添加一个drinkPotion()成员函数来应用药水的效果。

要求

- 根据游戏规则，画出main函数的流程图
- 根据游戏规则，画出格斗魔兽的流程图

- 根据类的设计，写出Creature.h/cpp, Player.h/cpp, Monster.h/cpp, Potion.h/cpp 的实现
- 写出main函数的实现
- 参考：https://www.learncpp.com/cpp-tutorial/chapter-17-comprehensive-quiz/

题目3. BlackJack （选做）

https://en.wikipedia.org/wiki/Blackjack

https://www.learncpp.com/cpp-tutorial/chapter-13-comprehensive-quiz/

实习四、面向对象程序设计二

题目1. 设计Shape类及其派生类

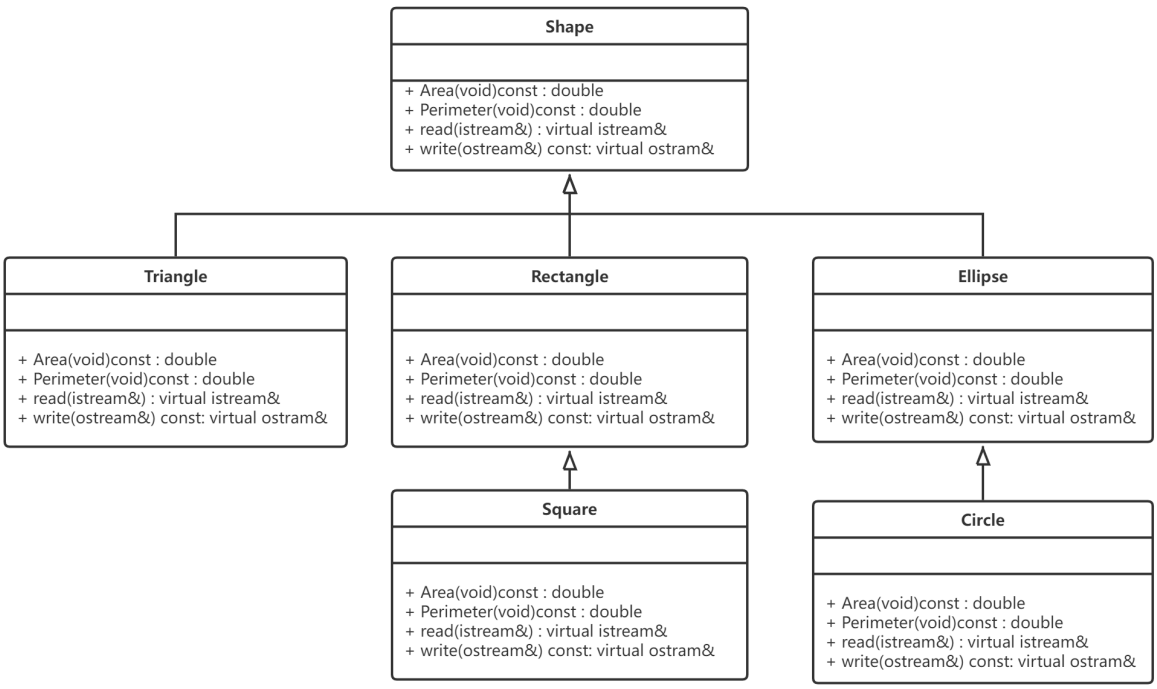
实现功能

- Area - 面积
- Perimeter - 周长
- operator<< 序列化
- operator>> 读取文件

  支持从数据中读取一行行的数据，创建形状图像，追加到对象列表中

```
//文件如下所示：
0, 3, 4, 5  //0-Triangle， 3,4,5 - 边长
1, 2, 3    //1-Rectangle， 2,3-长、宽
2, 5      //2-Square, 5-半径
3, 2      //3-Circle, 2-半径
4, 1, 3    //4-Ellipse, 1,3-长、短轴长
```

类的关系图如下



main() 代码

```
#include <iostream>
```

```cpp
#include "Triangle.h"
#include "Rectangle.h"
#include <list>

using namespace std;

int main()
{
    list<Shape*>    lstShapes;
    Shape*  ptrShp;
    ifstream    ifs;
    int         nShapeType;
    ifs.open("c:/shp.dat");

    while(!ifs.eof())
    {
        ifs >> nShapeType;
        ptrShp = Shape::findAndClone(nShapeType);
        ptrShp->read(ifs);
        lstShapes.push_back(ptrShp);
    }

    for (auto shp : lstShapes)
    {
        shp->write(cout) << ", ";
        cout << shp->area() << ", ";
        cout << shp->perimeter() << ", ";
        cout << endl;
    }

    for (auto shp : lstShapes)
    {
        delete shp;
    }
    return 0;
}

// static函数findAndClone 的实现示例
Shape* Shape::findAndClone(int nType)
{
    Shape* pshp = nullptr;
    switch(nType)
    {
    case 0: pshp = new Triangle(); break;
    case 1: pshp = new Rectangle(); break;
    ...
    default: pshp = nullptr; break;
    }
    return pshp;
}
```

题目2. 设计并实现矩阵类

设计并实现矩阵Matrix模板类，支持基本的数据运算。功能包括：

1. 支持多种数据类型，Matrix

2. 重载运算符+、-、*
3. 支持矩阵的<<, >>操作符

题目3. 词典查询

词典查询

读入字典文件，输入某个单词，输出中文释义 和 单词词性。

示例词典文件：下载词典dictionary.txt

五、中期大作业

训练项目要求

1. 任选一个训练项目，完成流程图绘制（或对象类图关系）、程序源代码、程序演示（视频或动画）和相关的报告文档。
2. 流程图或UML类图要规范、准确
3. 要求程序源代码不少于300行，尽量独立完成；若代码行超过600行的程序，可以由2人组团完成；
   代码超过1000行的程序，可以由3人组团完成。源代码需要有足够的注释，采用多文件结构（.h-声明，.cpp-实现）
4. 程序演示部分是程序的运行效果，以视频或动画呈现
5. 从设计、实现到运行情况的说明文档，格式PDF
6. 提交程序源代码、PDF文档和程序演示视频或动画文件
7. 截止日期：2023.5.10

训练项目选题

训练项目可以从https://codebus.cn/查看，或者其他网站自选。结合专业背景，建议从 图形与图像、游戏、算法 3个类别中选取。推荐项目：

1. 常见图像处理的演示
   推荐理由：可了解图像处理的算法和有趣的处理算法。
   推荐等级：★★
2. 直方图规定化+图像直方图统计+直方图均衡化
   推荐理由：可了解图像增强显示处理算法和效果。
   推荐等级：★★
3. 图像旋转
   推荐理由：可了解图像几何变换和重采样算法。
   推荐等级：★★
4. 素描算法
   推荐理由：可了解图像处理算法和直观的效果。
   推荐等级：★★
5. TIN 三角网的生成
   推荐理由：实现地形三维可视化，数字地面模型的表达。
   推荐等级：★★★
6. 数独辅助器
   推荐理由：实现数独游戏。
   推荐等级：★★★
7. 钟表模拟程序（表针形式）
   推荐理由：绘制钟表
   推荐等级：★

8. 音乐播放器
   推荐理由：自定义播放器
   推荐等级：★
9. 电子相册
   推荐理由：自定义电子相册
   推荐等级：★
10. 计算器
    推荐理由：运算功能。
    推荐等级：★★
11. 万年历
    推荐理由：图形化界面
    推荐等级：★
12. 植物大战僵尸之锤僵尸小游戏
    推荐理由：实现了锤僵尸、种植物、除草机、读档存档、收阳光等功能。
    推荐等级：★★★
13. 贪吃蛇
    推荐理由：这是一个传统的贪吃蛇游戏，基于链表实现
    推荐等级：★★
14. 扫雷（WinXP 扫雷的高仿版）
    推荐理由：高仿的扫雷游戏，游戏手感非常贴近原版。
    推荐等级：★★★
15. 拼图 2.0（by Redman）
    推荐理由：熟悉面向对象设计
    推荐等级：★
