

南开大学计算机大类本科生 2021—2022 学年第 2 学期
《高级语言程序设计 2-2》课程期末考试试卷（A 卷）

学院：_____专业：_____学号：_____姓名：_____成绩：_____

| |
|-----|
| 得 分 |
| |

一、单项选择（本题共 20 分，每小题 2 分）
请将答案填涂在答题纸的相应位置

1. 假设 `myClass` 是一个类，则执行“`myClass a(2), b[2], *p[2];`”语句时，调用的 `myClass` 的构造函数的次数为（ ）
A. 3 B. 4 C. 5 D. 6
2. 关于类的成员函数，描述不正确的是（ ）
A. 静态成员函数不但可以访问类的所有 `public` 数据成员，也可以访问类的所有 `private` 数据成员
B. 成员函数可以重载
C. 类的成员函数既可以是内联函数，也可以不是
D. 类的友元函数虽然可以访问类的私有数据成员，但友元函数并不是类的成员函数
3. 如果 `++` 和 `*` 都是友元方式重载的运算符，则表达式 `++i*k` 可以表示为（ ）
A. `operator*(i.operator++(), k)`
B. `operator*(operator++(i), k)`
C. `i.operator++().operator*(k)`
D. `k.operator*(operator++(i))`
4. 对于只在表的首、尾两端进行插入操作的线性表，宜采用的存储结构为（ ）
A. 顺序表（用数组存储的线性表）
B. 用头指针表示的单循环链表
C. 用尾指针表示的单循环链表
D. 单链表

5. 下列关于类的继承与派生的描述中，正确的是（ ）
- A. 公有继承中，对于基类中的所有成员，派生类的成员函数都可以直接访问
 - B. 派生类对象的地址可以赋值给指向基类的指针
 - C. 基类的友元同样是派生类的友元
 - D. 派生类隐式继承基类的构造函数和析构函数
6. 类 **B** 是类 **A** 的公有派生类，类 **A** 和类 **B** 中都定义了虚函数 **func()**，**p** 是一个指向类 **A** 对象的指针，则 **p->func()** 将（ ）
- A. 调用类 **A** 中的函数 **func()**；
 - B. 调用类 **B** 中的函数 **func()**
 - C. 根据 **p** 所指的对象的类型而确定调用类 **A** 或类 **B** 中的函数 **func()**
 - D. 既调用类 **A** 中函数，也调用类 **B** 中函数
7. 有如下函数模板定义：
- ```
template<class T>
T func(T x,T y)
{ return x*x+y*y; }
```
- 在下列对 **func** 的调用中，错误的是（ ）。
- A. **func(3, 5.5);**
  - B. **func(3.0, 5.5);**
  - C. **func(3, 5);**
  - D. **func<int>(3, 5.5)**
8. 类模板的模板参数可用做（ ）
- A. 数据成员的类型
  - B. 成员函数的类型
  - C. 成员函数的参数类型
  - D. 以上都可以
9. 下列关于关于虚函数的描述中，正确的是（ ）
- A. 虚函数是一个静态成员函数
  - B. 虚函数是一个非成员函数
  - C. 虚函数是静态联编的一种实现方式
  - D. 派生类的虚函数与基类中对应的虚函数具有相同的参数个数和类型
10. 当使用 **ifstream** 流类定义一个流对象并打开一个磁盘文件时，文件的默认打开方式为（ ）
- A. **ios::in**
  - B. **ios::in | ios::binary**
  - C. **ios::out**
  - D. **ios::in | ios::app**

|    |
|----|
| 得分 |
|    |

## 二、程序改错（本题共 16 分）

11. （8分）请找出程序中的 4 个语法错误，在答题纸上指出错误所在行，说明错误原因或进行改正。

```
[1] class Node{
[2] public:
[3] int data;
[4] Node *next;
[5] Node(int i) {
[6] data = i;
[7] next = nullptr;
[8] }
[9] }
[10] class List {
[11] private:
[12] Node *head;
[13] static int nodeNum;
[14] public:
[15] List()
[16] {
[17] head = new Node();
[18] }
[19] void Insert(int k, int data);
[20] };
[21] List::nodeNum = 0;
[22] void Insert(int k, int data) {
[23] Node *newNode = new Node(data);
[24] Node *p = head;
[25] for(int i = 1; i < k+1; i++)
[26] p = p->next;
[27] newNode->next = p;
[28] p->next = newNode;
[29] }
```

12. (8分) 请找出程序中的4个语法错误, 在答题纸上指出错误所在行, 说明错误原因或进行改正。

```
[1] #include <iostream>
[2] using namespace std;
[3] class A {
[4] protected:
[5] int x;
[6] public:
[7] A(int a) :x(a){}
[8] void f() { cout << "A" << endl; }
[9] };
[10] class B :public A {
[11] using A::A(int a);
[12] public:
[13] int y;
[14] B(int a, int b) :A() { y = b; }
[15] void f(int a) { cout << "B" << endl; }
[16] void showall() {
[17] cout << "x: " << x << endl;
[18] cout << "y " << y << endl;
[19] }
[20] int getx() {
[21] return x;
[22] }
[23] };
[24] int main() {
[25] B b1(10,20);
[26] cout << "x: " << b1.x << endl;
[27] A* a = &b1;
[28] a->f(10);
[29] A* a2 = new B(30,40);
[30] a2->f();
[31] }
```

|    |
|----|
| 得分 |
|    |

三、读程序写结果（本题共 24 分）

13. （6分）请在答题纸上写出以下程序在 Visual C++.Net 环境下的运行结果。

```
#include<iostream>
using namespace std;
class Student
{
 char *name;
 int age;
 float score;
 static int num;
 static float total;
public:
 Student(char *,int,float);
 void say() ;
 static float getAverage();
};
int Student::num=0;
float Student::total=0;
Student::Student(char *name,int age,float score)
{
 this->name=name;
 this->age=age;
 this->score=score;
 num++;
 total+=score;
}
void Student::say()
{cout<<name<<"的年龄是"<<age<<"，成绩是"<<score<<"（当前共"<<num<<"
名学生） "<<endl;}
float Student::getAverage()
{ return total/num; }
int main()
{
 (new Student("小明",15,90))->say();
 (new Student("李磊",16,80))->say();
 (new Student("张华",16,90))->say();
```

```

 (new Student("王康",14,60))->say();
 cout<<"平均成绩为"<<Student::getAverage()<<endl;
 return 0;
 }

```

14. (6分) 请在答题纸上写出以下程序在 Visual C++.Net 环境下的运行结果。

```

#include<iostream>
#include<cstring>
#include<stack>
using namespace std;
stack<int> s;
int main()
{
 int n=10;
 while(n)
 {
 s.push(n%2);
 n /= 2;
 }
 while(s.size())
 {
 cout<<s.top();
 s.pop();
 }
 return 0;
}

```

15. (6分) 请在答题纸上写出以下程序在 Visual C++.Net 环境下的运行结果。

```

#include<iostream>
using namespace std;
class BaseClass
{
public:
 BaseClass() {
 cout << "Construction of Base Class" << endl;
 }
 void Fun1() {
 cout << "Fun1() in BaseClass is called!" << endl;
 }
}

```

```

 }
 virtual void Fun2() {
 cout << "Fun2() in BaseClass is called!" << endl;
 }
};
class DerivedClass : public BaseClass
{
public:
 DerivedClass() { cout << "Construction of Derived Class" <<
endl; }
 void Fun1() { cout << "Fun1() in DerivedClass is called!" <<
endl; }
 void Fun2() { cout << "Fun2() in DerivedClass is called!" <<
endl; }
};
int main()
{
 DerivedClass d;
 BaseClass* pb = &d;
 pb->Fun1();
 pb->Fun2();
 return 0;
}

```

16. (6分)请在答题纸上写出以下程序在 Visual C++.Net 环境下的运行结果。

```

#include<iostream>
#include<string>
using namespace std;
class Animal
{
public:
 string name;
 Animal(string n) :name(n) {
 age = 10; cout << "Animal(string n) called" << endl;
 }
 Animal(string n, int a) :name(n), age(a) {
 cout << "Animal(string n,int a) called" << endl;
 }
 Animal() {
 name = "anonymity";age = 0; cout << "Animal() called" <<
endl;
 }
 ~Animal() { cout << "Animal is destructing" << endl; }
}

```



```

 void print() {
 cout << "name: " << name << endl;
 cout << "age: " << age << endl;
 }
protected:
 int age;

};

class Cat :public Animal
{
public:
 Animal animal;
 Cat(string name,string _hair) :Animal(name),hair(_hair)
 {
 cout << "Cat(string name,string _hair) called" << endl;
 }
 Cat(string name):animal(name)
 {
 cout << "Cat(string name) called" << endl;
 }
 Cat(){ hair = "short";cout << "Cat() called" << endl; }
 ~Cat() { cout << "Cat is destructing" << endl; }
 void print()
 {
 cout << "hair: " << hair << endl;
 }
protected:
 string hair;
};

int main()
{
 Cat p1("cat","short");
 p1.print();
 Animal* p2 = new Animal("dog", 10);
 p2 = &p1;
 p2->print();
 return 0;
}

```



|    |
|----|
| 得分 |
|    |

四、程序填空（本题共 20 分，每空 2 分）

17. 将一条链表上相邻的两个结点合并成一个结点，即将第 1 个结点与第 2 个结点合并，将第 3 个结点与第 4 个结点合并，.....，如果链表上结点个数为奇数，则最后一个结点不合并，直接作为合并后链表上的最后一个结点。合并 2 个结点的含义是：将两个结点的数据成员 **data** 值相加。链表结点的数据结构为：

```
struct node{
 int data;
 struct node* next;
};
```

以下函数的参数 **h** 指向待合并链表的首结点，请完善该函数，并填写在答题纸的相应编号处。

```
void merge(node *h)
{
 node *p1, *p2;
 if(h == NULL)
 return;
 p1 = h;
 p2 = _____(1)_____;
 while(p2)
 {
 p1->data += p2->data;
 p1->next = p2->next;
 _____(2)_____;
 p1=p1->next;
 if(p1 && p1->next)
 p2 = p1->next;
 else
 p2 = _____(3)_____;
 }
}
```

18. 以下程序定义了一个分数类 **Rational**，分子为 **nume**，分母为 **deno**。在类中重载了分数的+和-运算。请完善该程序，并填写在答题纸的相应编号处。

```
#include<iostream>
```

```

class Rational
{
 int nume,deno;
public:
 Rational(int x=0,int y=1){nume=x;deno=y;}
 void print();
 _____(1)_____
 _____(2)_____
};

Rational Rational::operator+(Rational a)
{
 Rational r;
 r.deno=a.deno*deno;
 r.nume=a.nume*deno+a.deno*nume;
 return r;
}

Rational operator-(Rational a,Rational b)
{
 Rational r;
 r.deno=a.deno*b.deno;
 r.nume=a.nume*b.deno-a.deno*b.nume;
 return r;
}

```

19. 以下程序定义了一个名为 **circle** 的类，用于表示一个圆形，该类的属性 **r** 表示圆的半径，**area** 函数用于计算圆的面积。**cylinder** 类为 **circle** 类派生出的圆柱体类，用于表示一个圆柱体，其中属性 **h** 表示圆柱体的高，基类 **circle** 表示底圆，**area** 函数用于计算圆柱体的面积。请完善该程序，并填写在答题纸的相应编号处。

```

#include<iostream>
using namespace std;
const double Pi = 3.14;
class circle {
public:
 circle(double R);
 double area()
 {
 return Pi*r*r;
 }
protected:

```

```

 double r;
 };

 _____(1)_____ {
 r = R;
 }

class cylinder :public circle
{
protected:
 double h;
public:
 _____(2)_____{};
 double area()
 {
 return _____(3)_____;
 }
};

void main() {
 cylinder c(1,2);
 cout << "The area of the cylinder c is "<< c.area()<< endl;
}

```

20. 下列程序将结构体变量 `t` 中的内容写入 `date.txt` 文件。请完善该程序，并填写在答题纸的相应编号处。

```

#include <fstream.h>
struct date
{
 int year,month,day;
};

int main()
{
 date t={2002,2,12};
 _____(1)_____;
 if (!outdate)
 {
 cerr << "\n 文件不能打开" << endl ;
 return;
 }
 _____(2)_____;
}

```

|    |
|----|
| 得分 |
|    |

## 五、程序设计（本题共 20 分）

### 21. （10 分）

去除一个已有的 C++ 源程序文件中的所有的 `for` 语句。要求：

- （1）在程序中由用户输入所要处理的 `cpp` 文件的文件名。
- （2）如果该文件不存在返回一个提示信息
- （3）去除程序中所有 `for` 语句后，得到的新程序存储到一个新文件中，新文件的文件名是 `new+源文件名`，例如原文件名是 `CPPfile.cpp`，新文件是 `newCPPfile.cpp`。

### 22. （10 分）

为了遏制大型互联网平台利用行业优势地位，进行大数据杀熟、算法歧视、诱导沉迷等不当行为，2022 年 3 月 1 日《互联网信息服务算法推荐管理规定》开始施行，《规定》明确，算法推荐服务提供者应当通过互联网信息服务算法备案系统填报服务提供者名称、服务形式、应用领域、算法类型等备案信息。2022 年 8 月，国内主流互联网服务提供商，包括抖音、腾讯、阿里、网易、微博等都将自己的核心推荐算法做了备案。下面是来自抖音的推荐算法备案信息：

算法运行机制：抖音个性化推送算法主要是基于用户历史的点击、时长、点赞、评论、分享、转发、不喜欢等行为数据，通过深度学习技术框架建立模型，预估用户对某个内容产生互动的概率，针对预估内容使用排序、打散、干预等机制和策略后，再向用户进行推荐。用户行为参考<用户，内容，互动>三个维度作为样本进入机器学习模型里训练，训练的结果用于更新用户模型和推荐新的内容。为了避免“信息茧房”问题的出现，抖音个性化推荐算法专门设计了“兴趣探索”机制。一方面每次推荐都会选择用户过去不常观看的内容类目进行一定比例的推荐。另一方面每次获取推荐内容的过程中会特别增加一条随机内容来保障用户可见内容的多样性。

以上是题目背景，以下是题目要求。

假设名为 `User_Video` 的类记录某个用户对某个视频的属性，`User_Video` 至少包括以下私有数据成员，例如，用户 ID (`User_ID`)、视频 ID (`Video_ID`)、是否点击、观看时长、是否点赞、是否评论、是否分享、是否转发、是否喜欢等。假设现在有 100 个用户，100 个视频。有一个数组，`long order[100][100]` 记录了基于抖音深度学习框架得到的针对每一个用户的推荐视频顺序。例如 `order[0]` 记录对第 0 号用户 100 个视频的推荐顺序，`order[0][0]` 是推荐给 0 号用户的首个视频的 ID，依此类推。`User_Video record[10000]` 记录每个用户针对每个视频的属性。第 0

号用户对 100 个视频的属性记录在 `record[0]-record[99]`，依此类推，题目要求：

(1) 设计一个名为 **Exploration** 的函数，请自己设计实现一个合理的“兴趣探索”机制。要求在 `order` 为每个用户推荐的前 10 个视频基础上，设计某种方法，插入 3 个用户过去不常观看的视频和 1 条随机选择的视频，并与原有的 6 个视频一起重新生成 **Top 10** 推荐视频。同时，简单解释你的方法的合理性。

(2) 开放性问题：目前，为了实现流量变现，抖音开辟了直播带货、达人探店等商业模式，请思考两方面问题：第一，作为一个用户，你如何发现抖音推荐算法出现了大数据杀熟、算法歧视、诱导沉迷、卖流量等不当行为，第二，如果你是抖音运营商，如何在遵循法律和道德的前提下，实现自身的经济利益。