C++程序设计试卷参考答案及评分标准

一、单项选择题（每题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
A	D	C	C	B	D	C	D	B	D

二、阅读程序,写出运行结果（每 6 分，共 30 分）

1.

1234　// 2 分，很简单

1369　// 3 分，错一处扣一分

说明：含有逻辑 Bug 的卷积实现，主要考察二维数组+双重循环。

2.

2023　// 2 分

Jpv　// 3 分，比较字符，拼合字符串

考查字符串数组操作

3.

2023

12

2133

23

43

每行 1 分，考查变量的类型和作用范围

4.

2c62

每个 1 分，格式 1 分。考查宏、回调函数

5.

33

32

34

31

每行有错扣 1 分，所有都不对扣 5 分。考查面向对象非继承部分知识。

6.

Circle Area 2

Point::print

Circle Area 2

Circle::print //每行 1 分

Circle::dtor

Point::dtor　// 两行 1 分

考查继承、多态

三、编写程序（10 分）

```
#include <iostream>
#include <cmath>
#define JINGDU 1e-5
#define PI 3.1415926
using namespace std;
unsigned long int factorial(unsigned long int n)
{
	unsigned long int i, m = 1;
	for(i = 0; i <= n; i++)
	{
		if(i != 0)	m = m * i;
	}
	return m;
}
double fcos(double x)
{
	double temp = 0.0, t = 5;
	int i = 0;
	while(x >= 2 * PI) x = x - 2 * PI;		//最多只有 2π 个弧度
	while(t >= jingdu)
	{
		t = pow(x, 2 * i) / factorial(2 * i);
		temp += pow(-1, i) * t;
		i++;
	}
	return temp;
}
int main()
{
	double r, temp;
	cout << "请输入弧度值，180°对应 3.1415926 弧度，以此类推...\n";
	cout << "弧度(rad):";
	cin >> r;
	temp = fcos(r);
	cout << "%.5lf\n", temp;
	cout << "%.51lf\n", cos(r); //用系统函数进行对比
	return 0;
}
```

四、编写程序（10 分）

```
#include <iostream>
using namespace std;
int main()
{
```

```
    char fu;
    double x, y, z;
    cout << "输入一个算符运算和两个浮点数：";
    cin >> fu >> x >> y;
    if (fu == '/') {  z = x / y; }
    else if (fu == '*') {z = x * y;  }
    else if (fu == '-') {z = x - y;   }
    else if (fu == '+') {z = x + y;  }
    else {cout << "非法运算符，无法参与运算！"; }
    cout << x << fu << y << "=" << z << endl;
    return 0;
}
```

五、编写程序（10 分）

```
#include <iostream>
using namespace std;
class Rational
{
public:
    Rational() { x=0; y=1; }
    Rational(int x, int y) { this->x=x; this->y=y; }
    Rational operator+(const Rational & p);
    Rational operator -() const;
    void Display() { cout << x << "/" << y << endl; }
private: int x, y;
};
Rational Rational::operator+(const Rational & p) // 重载加法运算
{
    int newx = x*p.y + y*p.x;
    int newy = y*p.y;
    return Rational(newx, newy);
}
Rational Rational::operator -() const
{return Rational(-this->x, this->y);}
```

六、编写程序（10 分）

```
#include <iostream>
#include <string>
using namespace std;
class Person // 父类
{
public:
    Person(const string & name, int age ){this->name = name; this->age = age;}
    virtual void show()=0; // 纯虚函数
protected:
    string name; // 姓名
```

```
    int age; // 年龄
};
class Teacher: public Person // 派生类
{
public:
    Teacher(const string & name, int age, const string & title): Person(name, age)
    {this->title = title;}
    void show() // 纯虚函数在派生类中要有定义，即实例化
    {cout << "Name: " << name << ", age: " << age << ",title: " << title << endl;}
private: string title; // 职称
};
class Student : public Person // 派生类
{
public:
    Student(const string & name, int age, int stuid) : Person(name, age)
    { this->stuid = stuid;}
    void show() // 每个派生类中都要实例化，否则该派生类仍然是抽象类
    {cout << "Name: " << name << ",age: " << age << ",stuid: " << stuid << endl;}
private:
    int stuid; // 学号
};
```

七、编写程序（10 分）

```
#include <iostream>
using namespace std;
template <typename T>
class Point{
public:
    Point(T a, T b);
    T Getx();
    T Gety();
private:
    T x, y;
};
template <typename T>
Point<T>::Point(T a, T b)
{ x=a; y=b; }
template <typename T>
T Point<T>::Getx()
{ return x; }
template <typename T>
T Point<T>::Gety()
{ return y; }
```