学号
姓名
班级

装订线

淮阴工学院课程考试试卷

专业：**计算机类专业**　课程名称: C++程序设计(下)　学分：2　试卷编号(A)
课程编号：1319423　考试方式：　闭卷　考试时间：100　分钟
拟卷人(签字)：____　拟卷日期：　2018.06.12　审核人(签字)：________

得分统计表：

题 号	一	二	三	四	五	总 分
得 分						

得分	

一、选择题：（每题 2 分，共 20 分）

```
1. #include<iostream>
using namespace std;
class Pair {
      int m,n;
public:
      Pair(int j,int k):m(j),n(k){}
      int get() {return m;}
      int get() const {return m+n;}
};
int main() {
Pair a(3,5);
const Pair b(3,5);
cout<<a.get()<<b.get();
return 0;
}
```

执行这个程序的输出结果是___B_____。

A、33　　B、38　　C、83　　D、88

2. 有如下程序：

```
#include <iostream>
using namespace std;
class Base {
protected:
         Base() { cout<<'A'; }
         Base(char c){ cout<<c; }
};
class Derived: public Base {
public:
         Derived( char c ) { cout<<c; }
};
int main() {
         Derived d1('B');
         return 0;
}
```

执行这个程序的输出结果是___C_____。

A、B　　B、BA　　C、AB　　D、BB

3. 有如下类的定义：

```
class Account{
public:
    Account(): id(0), money(0.0) { }
    Account(int _id, double _money): id(_id), money(_money) { }
    void deposite(double _money) { money += _money; }
    void withdraw(double _money) { money -= _money; }
private:
    int id;
    double money;
};
```

若执行语句：Account ac(1001, 1000), bc[3], *pc; 则类 Account 的构造函数被调用的次数是（C　）

(A) 1 次　　(B) 3 次　　(C) 4 次　　(D) 5 次

4. 有如下类定义：

```
class MyBase {
      int k;
public:
      MyBase(int n=0):k(n) { }
      int value()const { return k;}
};
class MyDerived: MyBase {
      int j;
public:
      MyDerived(int i): j(i) {}
      int getK()const { return k; }
      int getJ()const { return j; }
};
```

编译时发现有一处语法错误，对这个错误最准确的描述是___A__什么

A、函数 getK 试图访问基类的私有成员变量 k

B、在类 MyDerived 的定义中，基类名 MyBase 前缺少关键字 public、protected 或 private

C、类 MyDerived 缺少一个无参的构造函数

D、类 MyDerived 的构造函数没有对基类数据成员 k 进行初始化

5. 下列运算符函数中肯定不属于类 FunNumber 的成员函数的是____D____。

A、int operator- (FunNumber);

B、FunNumber operator - ();

C、FunNumber operator - (int);

D、int operator - (FunNumber, FunNumber);

6. 有如下程序：

```
#include <iostream>
using namespace std;
class CD {
public:
      ~CD() { cout<<'C'; }
private:
      char name［80］;
};
int main() {
      CD a,*b, d[2];
      return 0;
}
```

执行这个程序的输出结果是____B____。

A、CCCC　　B、CCC　　C、CC　　D、C

7. 有如下程序：

```
#include <iostream>
using namespace std;
class MyClass {
public:
      MyClass() { ++count; }
      ~MyClass() { --count; }
      static int getCount() { return count; }
private:
       static int count;
};
int MyClass::count=0;
int main()
{
      MyClass obj;
      cout<<obj.getCount();
      MyClass *ptr=new MyClass;
      cout<<MyClass::getCount();
      delete ptr;
      cout<<MyClass::getCount();
      return 0;
}
```

执行这个程序的输出结果是____A____。

A、121

B、232

C、221

D、122

8. 有如下类的定义：

```
class Base{
public:
    int a;
protected:
    int b;
private:
    int c;
};
class Derived: protected Base{
public:
    int m;
protected:
    int n;
};
```

则派生类 Derived 中保护数据成员的个数是（　C　）

(A) 1 个　　(B) 2 个　　(C) 3 个　　(D) 4 个

9. 有如下类的定义：

```
class Point{
public:
    Point(double _x = 0.0, double _y = 0.0): x(_x), y(_y) { }
    double& getX() { return x; }
    double& getY() { return y; }
protected:
    double x, y;
};
class Point3D: public Point{
public:
    Point3D(double _x = 0.0, double _y= 0.0, double _z = 0.0):____________, z(_z) { }
    double& getZ() { return z; }
protected:
    double z;
};
```

学号　　线

姓名　　订

班级　　装

则下列关于以上横线位置处对基类数据成员初始化的方式正确的是（　B　）

(A) x(_x), y(_y)　　　　(B) Point(_x, _y)

(C) Point::x(_x), Point::y(_y)　　　　(D) x = _x, y = _y

10. 有如下程序代码：

```
#include <iostream>
using namespace std;
class Square{
public:
    Square(double _width = 0.0): width(_width) { }
    virtual double area() const { return width*width; }
protected:
    double width;
};
class Cube: public Square{
public:
    Cube(double _width): Square(_width) { }
    double area() const { return width*width*width; }
};
int main(){
    Cube a(10);
    Square &b = a;
    Square c = a;
    cout<<b.area()<<c.area()<<endl;
    return 0;
}
```

则程序执行后的输出结果为（　C　）

(A) 100100　　(B) 10001000　　(C) 1000100　　(D) 1001000

得分	

二、程序阅读题：（每题 10 分，共 40 分）

1. 阅读以下程序：

```
#include <iostream>
#include <string>
using namespace std;
class Book
{
public:
Book(string _name = "", int _bnum = 0): bnum(_bnum)
{
      total += bnum;
}
~Book()
{
      total -= bnum;
}
void borrow(int num)
{
      if(bnum>num)
      {
            bnum -= num;
            total -= num;
      }
}
void back(int num)
{
      bnum += num;
      total += num;
}
void print() const
{
      cout<<bnum<<" "<<total<<" ";
}
private:
      string name;
      int bnum;
      static int total;
};
int Book::total = 0;
int main()
{
      Book book1("C++", 10);
      Book book2("Java", 20);
      book1.borrow(5);
      book2.back(2);
      book1.print();
      book2.print();
      return 0;
}
```

程序执行后的输出结果为__5　27　22　27_______

学号

姓名

班级

装 订 线

2. 阅读以下程序：

```
#include<iostream>
using namespace std;
class Tdate
{
private:
  int year;
  int month;
  int day;
public:
  void setDate(int a,int b,int c);
  int getYear();
  void print();
  bool isLeapYear();
};
void Tdate::setDate(int a,int b,int c)
  {
        year=a;
        month=b;
        day=c;
  }
int Tdate::getYear()
{
  return this->year;
}
void Tdate::print()
{
  cout<<"Today is:"<<year<<"/"<<month<<"/"<<day<<endl;
}

bool Tdate::isLeapYear()
{
  bool leapYear=false;
  if((year%4==0&&year%100!=0)||year%400==0)
  {
        leapYear=true;
  }
  return leapYear;
}
int main()
{
  Tdate today;
  today.setDate(2018,6,14);
  today.print();
  if(today.isLeapYear()==true)
        cout<<today.getYear()<<" is leap year!"<<endl;
  else
        cout<<today.getYear()<<" is not leap year!"<<endl;
  return 0;
}
```

程序执行后的输出结果为____Today is_2018/6/14

_____2018 is not leap year______________

3. 阅读以下程序：

```
#include <iostream>
using namespace std;
class Sales
{
public:
    Sales(double _price, int _amount): price(_price), amount(_amount), samount(0) { }
    void sell(int num)
    {
        if(amount>num)
        {
            amount -= num;
            samount += num;
        }
    }
    virtual double income() const
    {
        return price*samount;
    }
protected:
    double price;
    int amount;
    int samount;
};
class NewSales: public Sales
{
public:
    NewSales(double _price, int _amount, double _discount):
```

学号 姓名 班级

装 订 线

```
            Sales(_price, _amount), discount(_discount) { }
    double income() const
    {
        return price*samount*discount;
    }
protected:
    double discount;
};
int main()
{
    Sales sa1(10, 1000);
    NewSales sa2(20, 2000, 0.9);
    sa1.sell(20);
    sa2.sell(30);
    Sales &sa = sa2;
    cout<<sa1.income()<<" "<<sa.income()<<endl;
    return 0;
}
```

程序执行后的输出结果为____200　　540__________________

4. 阅读以下程序：

```
#include <iostream>
using namespace std;
class complex
{
private:
  double real;
  double image;
  const double pie;
public:
  //complex(void);
  complex(double re=0,double im=0);
  complex (const complex & cp);
  ~complex(void);
  double getReal()const;//
  double getImage()const;//
  void updateComplex(double rea,double ima);
void operator=(complex &a);
  complex operator+(complex &cp);
  complex operator++();//前++
  complex operator++(int);//后++
  complex operator--();//前--
  complex operator--(int);//后—
};
complex::complex(double re,double im):pie(3.1415926)
{
  this->real=re;
  this->image=im;
}
complex::complex(const complex &cp):pie(3.1415926)
{
  this->real=cp.real;
  this->image=cp.image;
}
void complex::updateComplex(double a,double b)
{
  real=a;
  image=b;
}
double complex::getReal()const
{
  return real;
}
double complex::getImage()const
{
  return image;
}
complex complex::operator+(complex &cp)
{
  complex result(this->real+cp.real,image+cp.image);
  return result;
}
void complex::operator=(complex &cp)
{
  real=cp.real;
  image=cp.image;
}
complex complex::operator++()
{
```

学号　　姓名　　班级

装　订　线

```
  real=real+1;
  image=image+1;
  return *this;
}
complex complex::operator++(int)
{
  complex temp=*this;
  real=real+1;
  image=image+1;
  return temp;
}
complex::~complex(void)
{
}
ostream& operator<<(ostream& oo,const complex &cp)
{
  oo<<cp.getReal()<<"+"<<cp.getImage()<<"i"<<endl;
  return oo;
}
istream& operator>>(istream& in,complex &cp)
{
  double _real,_image;
  in>>_real>>_image;
  cp.updateComplex(_real,_image);
  return in;
}
int main()
{
  complex c1(1,1);
  complex c2(2,3);
  cout<<c1+c2;
  c1.updateComplex(5,6);
  cout<<c1++;
  cout<<++c2;
}
```

程序执行后的输出结果为____3+4i_
____5+6i_
____3+4i_______

得分	

三、编写程序题（每题 20 分，共 40 分）。

第一题：定义矩形类(rectangle)，它宽而高（宽，高）。当描述该类的对象时，如果给定两个参数，则分别将它们设置为宽度和高度；如果只给出一个参数，则宽度和高度相同（即正方形）。同时，编写了一个可以计算并输出面积的成员函数（printArea）。编写相应的调试功能，测试类的正确性。

第二题：定义一个抽象类 Shape,其中有纯虚函数 double area()；由 Shape 再派生出 Circle、Rectangle 类， Circle 类中有中心点坐标为 int x,y 半径为 double R，面积为 double area()；Rectangle 类有长（long）与宽（wide），面积为 double area()，设计一个函数 dispalyArea()，其功能为显示面积，其形参为 Shape 类指针，请编程实现动态联编计算面积的多态性。