面向对象(C++)程序设计（上机考试）

样题 1．下列 Shape 类是一个表示形状的抽象类，Area()为求图形面积的函数，Total()则是一个通用的用以求不同形状的图形面积总和函数。请从 Shape 类派生三角形类(triangle)、矩形类(rectangle),并给出具体的求面积函数。编写程序验证求面积函数的正确性。Shape、 total 的定义如下所示。

```
Class shape{
Pubilc:
Virtual float area()=0
};
float total (shape *s[ ], int n)
{
float sum=0.0;
for(int i=0; i<n; i++)
sum += s [ I ] ->area ();
return sum;
}
```

解答: #include <iostream.h>

```
class shape{
    public:
        virtual float area()=0;
    };
float total(shape *s[], int n)
{
    float sum=0;
    for(int i=0; i<n; i++)
        sum += s[i] -> area();
    return sum;
}
class triangle : public shape{
    protected:
        float H, W;
    public:
        triangle(float h, float w) { H=h; W=w;}
        float area() { return H*W*0.5;}
};
class rectangle : public triangle{
    public:
        rectangle(float h, float w) : triangle(h, w) {}
        float area() { return H*W;}
};
void main()
{
    shape *s[4];
    s[0] = new triangle( 3.0, 4.0 );
    s[1] = new rectangle( 2.0, 4.0 );
    s[2] = new triangle( 5.0, 8.0 );
    s[3] = new rectangle( 6.0, 8.0 );
    float sum = total(s,4);
    cout << "The total area is:" << sum << endl;
}
```

样题 2．以面向对象的概念设计一个类,此类包括 3 个私有数据，unlead(无铅汽油), lead 有铅汽油, total (当天总收入)。其中，无铅汽油价格是¥17/升，有铅汽油价格是¥16/升，请以构造函数的方式建立此值，并编写程序，该程序能够根据加油量，自动计算出油站当天的总收入。

解答: #include <iostream.h>

```
class income {
    private:
        float unlead, lead, total;
    public:
        income(float ul, float l){unlead=ul; lead=l; total=0.0;}
        float calculate(float, float);
};
float income :: calculate(float unleadcontent, float leadcontent)
{
    total = unlead*unleadcontent + lead*leadcontent;
    return total;
}
void main()
{
    float unleadcontent, leadcontent, total;
    income account(17, 16);
    cout << "Please input unlead content:" << endl;
    cin >> unleadcontent;
    cout << "Please input lead content:" << endl;
    cin >> leadcontent;
```

```
   total = account.calculate(unleadcontent, leadcontent);
   cout << "The total income is:" << total << endl;
}
```

样题 3． 编写一个计算两个给定长方形的面积的程序，要求长方形用一个类(Rectangle)来表示，在该类中增加定义一个成员函数 add_area(), 该成员函数使用对象作为参数，用来计算两个给定长方形的面积。

解答:

```
#include <iostream.h>

class Rectangle {
   private:
      float H, W;
   public:
      Rectangle(float h, float w){H=h; W=w;}
      float area(){ return H*W; }
      float add_area(Rectangle &);
};
float Rectangle :: add_area(Rectangle &Rec)
{
   return area() + Rec.area();
}
void main()
{
   float h1, w1, h2, w2, totalarea;
   cout << "Please input the 1st rectangle H & W:" << endl;
   cin >> h1 >> w1;
   Rectangle rec1(h1,w1);
   cout << "Please input the 2st rectangle H & W:" << endl;
   cin >> h2 >> w2;
   Rectangle rec2(h2,w2);
   totalarea = rec1.add_area(rec2);
   cout << "The total area of the two rectangle is:" << totalarea << endl;
}
```

样题 4. 定义一个复数类 COMPLEX, 该类至少提供加、减、赋值、输出等操作，所有操作均以友元形式实现。编写程序验证其功能。

解答:

```
#include <iostream.h>

class complex{
   private:
      float real;
      float imag;
   public:
      complex(){};
      complex(int r, int i) {real=r; imag=i;}
      friend complex operator + (complex &, complex &);
      friend complex operator - (complex &, complex &);
      void show();
};
complex operator + (complex & a, complex & b)
{
   float r, i;
   r=a.real+b.real;
   i=a.imag+b.imag;
   return complex(r,i);
}
complex operator - (complex & a, complex & b)
{
   float r, i;
   r=a.real-b.real;
   i=a.imag-b.imag;
   return complex(r,i);
}
void complex :: show()
{
   if (imag>0)
      {
         if (imag == 1)
     cout << real << "+i";
         else  cout << real << "+" << imag << "i";
      }
   else if (imag<0)
      {
         if (imag == -1)
     cout << real << "-i";
         else  cout << real << imag <<
"i";
        }
    else    cout << real;
 }
 void main()
 {
    complex a(4,5), b(2,3), x, y;
    x=a+b;
    y=a-b;
    a.show();
    cout << "+";
    b.show();
    cout << "=";
    x.show();
    cout << endl;
    a.show();
    cout << "-";
    b.show();
    cout << "=";
    y.show();
    cout << endl;
 }
```

样题 5. 定义一个平面几何中点的位置类 POSITION，它应该包含有移动、计算两点间的距离(包括到原点的距离)，求 X 坐标、Y 坐标等操作，其中计算两点间的距离以友元函数形式实现。编写程序验证其功能。

解答:

```
 #include <iostream.h>

 #include <math.h>
 class Position{
    private:
        float X, Y;
    public:
        Position(float xi, float yi){ X=xi,
Y=yi;}
        void Move(float xo, float
yo){ X+=xo, Y+=yo;}
        float GetX(){return X;}
        float GetY(){return Y;}
        float distanceToOrigin();
        friend float distance(Position &,
Position &);
 };
 float Position :: distanceToOrigin()
 {
    return sqrt( X*X + Y*Y );
 }
 float distance(Position & a, Position & b)
 {
    float dx = a.X - b.X;
    float dy = a.Y - b.Y;
    return sqrt( dx*dx + dy*dy);
 }
 void main()
 {
    Position p1(1.5, 3.5), p2(4.5, 6.5);
    p1.Move(3.5, 5.5);
    float dis0 = p1.distanceToOrigin();
    float dis = distance( p1, p2 );
    cout << "The distance p1(" <<
p1.GetX() << "," << p1.GetY()
            << ") to origin is: " << dis0 <<
endl;
    cout << "The distance between p1(" <<
p1.GetX() << "," << p1.GetY()
            << ") and p2(" << p2.GetX() <<
"," << p2.GetY() << ") is "
            << dis << endl;
 }
```

样题 6. 利用类和对象，编制出一个卖瓜的程序。每卖一个瓜要计出该瓜的重量，还要计算所卖出瓜的总重量及总个数，同时卖瓜时还允许退瓜。(提示:将每个瓜设为对象;用静态成员变量分别统计卖出瓜的总重量和总个数；卖瓜行为用构造函数模拟，退瓜行为用析构函数模拟。)

解答:

```
 #include <iostream.h>

 class Watermelon{
    private:
        static int n;
        static float totalWeight;
        float weight;
    public:
        Watermelon(float w)
        {
            n++;
            weight=w;
            totalWeight += w;
        }
    Watermelon(Watermelon & wa)
    {
      n++;
      weight=wa.weight;
      totalWeight += weight;
    }
    ~Watermelon()
    {
      n--;
      totalWeight -= weight;
    }
    int getWeight()
    {
      return weight;
    }
    static int getNum()
    {
      return n;
    }
    static int getTotal()
    {
      return totalWeight;
    }
};
int Watermelon :: n = 0;
float Watermelon :: totalWeight = 0;
void main()
{
  float w;
  cout << "The initial weight of
watermelon: "
       << Watermelon :: getTotal() <<
endl;
  cout << "Please input weight of
watermelon:" << endl;
  cin >> w;
  Watermelon wa1(w);
  cout << "Please input weight of
watermelon:" << endl;
  cin >> w;
  Watermelon wa2(w);
  cout << "Please input weight of
watermelon:" << endl;
  cin >> w;
  Watermelon wa3(w);
  cout << Watermelon :: getNum() <<
"Nos. watermelons were sold." << endl;
  cout << "The total weight is: " <<
Watermelon :: getTotal() << endl;
  wa3.Watermelon::~Watermelon();
  cout << "Now one watermelon was
withdrawed!" <<endl;
  cout << Watermelon :: getNum() <<
"Nos. watermelons were sold." << endl;
  cout << "The total weight is: " <<
Watermelon :: getTotal() << endl;
}
```

样题 6. 编写一个程序，用于计算三角形、矩形和圆的总面积。(提示：由于尚不能确定该程序计算的具体形状，可以先定义一个抽象的类 shape, 对于具体种类的形状，通过从 shape 派生一个类来对其进行描述。)

解答:

```
#include <iostream.h>

class shape{
  public:
    virtual float area()=0;
  };
float total(shape *s[], int n)
{
  float sum=0;
  for(int i=0; i<n; i++)
    sum += s[i] -> area();
  return sum;
}
class triangle : public shape{
  protected:
    float H, W;
  public:
    triangle(float h, float w) { H=h;
W=w;}
    float area() { return H*W*0.5;}
};
class rectangle : public triangle{
  public:
    rectangle(float h, float w) : triangle(h,
w) {}
    float area() { return H*W;}
};
class circle : public shape{
  protected:
    float radius;
  public:
    circle(float r) { radius=r; }
    float area() { return radius*radius*3.14; }
};

void main()
{
  shape *s[4];
  s[0] = new triangle( 3.0, 4.0 );
  s[1] = new rectangle( 2.0, 4.0 );
  s[2] = new circle( 5.0 );
  s[3] = new circle( 8.0 );
  float sum = total(s,4);
  cout << "The total area is:" << sum << endl;
}
```

题样 7. 编写一个程序，可以输入 3 个学生的总分，并按总分从高到低排序，要求设计一个学生类 STUDENT，并编写其所有成员函数的代码，类 STUDENT 的定义如下：

```
class STUDENT {
     int total;              // 总分成绩
   pubic:
     void get_score();       //获取一个学生的成绩
     void display();         //显示一个学生的成绩
     void sort (STUDENT *);  //将若干学生成绩按总分从高到低排序
   }
```

解答:

```
#include <iostream.h>

#include <iomanip.h>
class Student{
    int total;
  public:
    Student(int m){total=m;}
    int get_score();
    void display();
    void sort(Student *s[3]);
};
int Student :: get_score()
{
  return total;
}
void Student :: display()
{
  cout << setw(6) << total << endl;
}
void Student :: sort(Student * s[3])
{
  Student * t;
  for (int i=0; i<3; i++)
  {
    for (int j=2; j>i; j--)
    {
      if (s[j]->get_score() > s[j-1]->get_score())
      {
     t=s[j];
     s[j]=s[j-1];
     s[j-1]=t;
      }
    }
  }
  cout << "The sorted score is: ";
  for (int k=0; k<3; k++)
    cout << setw(6) << s[k]->get_score();
  cout << endl;
}
void main()
{
  int s0, s1, s2;
  cout << "Please input 3 score:" << endl;
  cin>>s0>>s1>>s2;
  Student *s[3];
  s[0]=new Student(s0);
  s[1]=new Student(s1);
  s[2]=new Student(s2);
  s[0]->sort(s);
}
```

样题 8. 设计一个栈操作类，该类包含入和出栈成员函数，编写程序，入栈一组数据：(5, 2, 6, 7, 3)，然后屏幕显示出栈结果。

解答:

```
#include <iostream.h>
```

```
#include <iomanip.h>
const int size=20;
class stack{
   private:
      int data[size];
      int top;
   public:
      stack(){ top=-1; }
      void push(int);
      int pop();
};
void stack :: push( int c)
{
   if (top>=19)
      cout << "stack overflow!" << endl;
   else
      data[++top]=c;
}
int stack :: pop()
{
   if (top<=-1)
   {
      cout << "stack underflow!" << endl;
      return NULL;
   }
   else
      return data[top--];
}

void main()
{
   stack s;
   s.push(5);
   s.push(2);
   s.push(6);
   s.push(7);
   s.push(3);
   for (int i=0; i<5; i++)
      cout << setw(6) << s.pop();
   cout << endl;
}
```

样题 9. 编写一个输入、显示学生(用类 student 表示)和教师(用类 teacher 表示)数据的程序，学生的数据包括：编号(NO)、 姓名(NAME)和班号 (CLASS)， 教师的数据包括：编号(NO)、 姓名(NAME)和部门(DEPT)。要求将编号、姓名的输入和显示两项操作设计成一个类 person，并将它作为 student 类和 teacher 类的基类。

解答:

```
#include <iostream.h>

#include <string.h>

class person{
   protected:
      char No[5];
      char Name[10];
   public:
      virtual void input();
      virtual void show();
};
class student : public person{
   private:
      char Class[3];
   public:
      virtual void input();
      virtual void show();
};
class teacher : public person{
   private:
      char Dept[3];
   public:
      virtual void input();
      virtual void show();
};

void person :: input()
{
   cout << "Please input NO.:" << endl;
   cin >> No;
   cout << "Please input NAME:" <<
endl;
   cin >> Name;
}

void person :: show()
{
   cout.setf(ios::left);
   cout.width(8);
   cout << "NO.:" << No << endl;
   cout.width(8);
   cout << "NAME:" << Name << endl;
```

```
 }


 void student :: input()
 {
    person :: input();
    cout << "Please input CLASS:" <<
endl;
    cin >> Class;
 }

 void student :: show()
 {
    person :: show();
    cout.setf(ios::left);
    cout.width(8);
    cout << "CLASS:" << Class << endl;
 }

 void teacher :: input()
 {
    person :: input();
    cout << "Please input DEPT:" << endl;
    cin >> Dept;
 }

 void teacher :: show()
 {
    person :: show();
    cout.setf(ios::left);
    cout.width(8);
    cout << "DEPT:" << Dept << endl;
 }

 void main()
 {
    student stu1, stu2;
    teacher tea1;
    cout << "Please input 1st student's
information:" << endl;
    stu1.input();
    cout << "Please input 2st student's
information:" << endl;
    stu2.input();
    cout << "Please input teacher's
information:" << endl;
    tea1.input();
    cout << endl << "STUDENT:" << endl;
    stu1.show();
    stu2.show();
    cout << endl << "TEACHER:" <<
 endl;
    tea1.show();
 }
```

样题 10. 编写一个可以删除文本文件中所有以“//”开头字符串的 C++程序。要求必须使用 C++的 I/O 流成员函数来完成。

解答:

```
#include <iostream.h>

#include <fstream.h>

void main(int argc, char * argv[])
{
   char ch;
   if (argc!=3){
      cout << "Error, You shall use this
programs as: \n\t"
       << "purgefile filename1 filename2"
<< endl;
      return;
   }
   ifstream myin(argv[1]);
   if (!myin){
      cout << "Can't open file " << argv[1]
<< endl;
      return;
   }
   ofstream myout(argv[2]);
   if(!myout){
      cout << "Can't create file " << argv[2]
<< endl;
      return;
   }
   while (myout && myin.get(ch)){
      if ( ch == '/' ){
         myin.get(ch);
         if ( ch == '/'){
     do{

     }  while  (myin.get(ch)  &&  ((ch!='
')&&(ch!='\n')));
         }
```

```
            else {
          myout.put('/');
          myout.put(ch);
          myin.get(ch);
          do{
          myout.put(ch);
          }  while  (myin.get(ch)  &&  ((ch  !='
')&&(ch!='\n')));
          if ((ch == ' ')||(ch == '\n')){
            myout.put(ch);
          }
            }
           }

           else if ((ch==' ')||(ch=='\n')) {
             myout.put(ch);
           }
           else {
             myout.put(ch);
             myin.get(ch);
             do{
             myout.put(ch);
             }  while  (myin.get(ch)  &&  ((ch!='
')&&(ch!='\n')));
             if ((ch == ' ')||(ch == '\n')) {
          myout.put(ch);
             }
           }
         }
         myin.close();
         myout.close();
         cout << "purge completed." << endl;
       }
```

样题 11. 已知类 test 含有整型私有数据成员 num，编写主程序，要求：

1) 写出 test 的完整定义，并含有必要的函数成员;

2) 建立长度为 9、元素类型为 Test 的动态数组且初值为 0;

3) 将各元素的值均设置为 0x7FF;

4) 显示各元素的值;

5) 删除动态数组;

解答:

```
#include <iostream.h>
#include <iomanip.h>
```

```
         class test{
           private:
             int num;
           public:
             test() { num=0; }
             test(int n) { num=n; }
             void SetNum(int n) { num=n; }
             int GetNum() { return num; }
         };

         void main()
         {
           test *ptr=new test[9];
           int i;
           for (i=0; i<9; i++)
             ptr[i].SetNum(0x7FF);
           for (i=0; i<9; i++)
             cout   <<   setw(8)   <<   hex   <<
ptr[i].GetNum();
           cout << endl;
           delete []ptr;
}
```