一.简答题 (共 10 题，50.0 分)

1. (简答题 5.0 分)

```
class WorkingHours{
   int todayHours;
   static int sum;
public:
        WorkingHours (int i = 0): todayHours(i) { sum += todayHours;}
        static int GetSum ( ){
                return sum;
         }
         int GetHours ( ){ return todayHours;}
};
int WorkingHours::sum = 0;
int main( )
{
         WorkingHours mon, tue(5), wed(8);
         cout<< WorkingHours:: GetSum ( )< <endl;
         cout< < mon.GetHours( )< <endl;
         cout< <tue.GetHours( )< <endl;
         cout< <wed.GetHours( )< <endl;
         return 0;
}
```

2. (简答题，5.0 分)

```
template<class T> //已知大写字母 A 的 ASCII 码值是 97
   class Swap{
            T val1, val2;
                public:
            Swap(T a, T b){ val1 = b; val2 = a;}
                void print( ){cout< <val1< <endl< <val2 < <endl;}
         };
int main ( )
{
          Swap<int> num('d', 'D');
      num.print( );
          return 0;
}
```

3. (简答题，5.0 分)

```
class Person{
        string name;
        int age;
public:
        Person(string n = " ", int c = 0): name(n),age(c) {cout <<"Constructor" < <endl;}
        Person(const Person& in): name(in.name), age(in.age) {cout<<"Copy constructor"<
<endl;}
        ~Person() {cout <<"Deconstructor" < <endl;}
        int GetAge() {return age;}
};
bool AgeCompare(Person a, Person& b)
{
         if(a.GetAge() > b.GetAge()) {cout<< "Elder" < <endl; return true;}
         else {cout< < "Younger" < <endl; return false;}
}
int main(){
         Person p1("张三",17), p2("李四",18);
        cout< <AgeCompare(p1, p2)< <endl;
         return 0;
}
```

4. (简答题，5.0 分)

```
class Complex{
        float real, imag;
public:
        Complex(float a, float b):real(a), imag(b){}
        friend Complex operator+ (Complex& c1, Complex& c2)
                {return Complex(c1.real + c2.real, c1.imag + c2.imag):}
        void Print( ){cout< <real<<" "< <imag< <endl;}
 };
int main( ){    Complex c1(4, -1), c2(2, 3);
        Complex* c3 = new Complex(c1 + c2);
        C3->Print( );
        delete c3;
        return 0;
 }
```

5. (简答题，5.0 分)

```
class Test{
        char lett;
public:
        Test (char v): lett(v){}
        Test (int v): lett(v){}
        Test (const Test& in): lett(in.lett){}
        Test operator++(int a)
        {
                Test temp = *this;
                lett+ +;
                return temp;
         }
         Test operator++( )
         {
                lett++;
                return *this;
          }
          void Print( ) {cout<< "Letter: " < <lett< <endl;}
 };
int main( ){
        Test v1('B'), v2(66);
        Test v3 = v1++;
        Test V4= ++v2;
        v1.Print( );
        v2.Print ( );
        v3.Print ( );
    v4.Print( );
    return 0;
}
```

6. (简答题, 5.0 分)

```
class Base{
        int a,b;
public:
        Base(int i, int j){ a = i;b = j ; cout<<"Base" < <endl;}
        Base(Base &T){ a = T.a ; b = T.b ; cout<<"Copy Base"< <end;}
        - Base( ) {cout<<"- Base" < <endl;}
        int Sum(){return a+b;}
        int GetA( ){return a;}
        int GetB( ){return b;}
};
class Derived: public Base{
      int c;
public:
        Derived( int i,int j, int k): Base(i,j), c(k){cout< < "Derived"< <endl; }
        ~Derived( ) {cout< <"~Derived" < <end;}
        int Sum (){ return c + GetA() + GetB( ); }
};
int main( ) {    Derived obj1(3, 4, 5);
    Base obj2(obj1);
      cout< <obj1.Sum( )<<" "<<obj2.Sum( )<<endl;
      return 0;
}
```

7. (简答题 5.0 分)

```
class A{
public:
        A( ){cout< <"A"< <endl;}
        ~A( ){cout<<"~A"<<endl;}
        virtual void Fun( ) {cout <<"AFun( )"< <endl;}
};
class B: public A{
public:
        B( ){cout<<"B"< <end;}
        - B( ){cout<<"-B"< <endl;}
        virtual void Fun( ) {cout << "BFun( )"< <endl;}
}:
int main( ){
        B obj2;
        A* ptr = &obj2;
        ptr->Fun( );
        return 0;
}
```

8. (简答题, 5.0 分)

```
class Part{
        int num;
          public:
          Part(int n): num(n) {cout<<"Part:" < <num< <endl;}
};
class Whole{
   Part one, two;
public:
          Whole(Part x, Part y): two(x), one(y) {cout< <"Whole" < <endl;}
};
int main( ) {
     Whole(2, 5);
            return 0;
}
```

9. (简答题, 5.0 分)

```
class A{
    string type;
public:
        A(string s):type(s) {cout< <type< <endl;}
        -A( ){cout<<"-A" < <endl;}
      virtual int num( ) = 0;
};
class B: public A{
    int b;
public:
        B(int i): A("A1"), b(i) {cout<<"B"<<endl;}
        ~B( ){cout<<"~B"<<endl;}
        int num( ) {return b*b;}
};
class C: public A{
        int c;
public:
        C(int j): A("A2"), c(j) {cout<<"C" < <endl;}
    ~C( ){cout<<"~C"<<endl;}
    int num( ) {return c*c;}
};
int main( ){
    C c1(3);
        A* aptr = &c1;
        cout< <aptr-> num( )< <endl;
        return 0;
}
```

10. (简答题, 5.0 分)

```
int main( ){
        fstream dataFile;
        dataFile.open("num.txt", ios::o
        dataFile<<"123456789";
        dataFile.close( );

        dataFile.open("num.txt", ios::in
        char ch1;
        dataFile.seekg (3L, ios::beg);
        dataFile.get(ch1);
        cout<<ch1<<endl;
        dataFile.seekg (-2L, ios::end);
        dataFile.get(ch1);
        cout<ch1 <<endl;
        dataFile.seekg(- 3L, ios::cur);
        dataFile.get(ch1);
        cout<<ch1 <<endl;
        cout<<dataFil.tellg( )<<endl;
        dataFile.close( );
        return 0;
}
```

二.其它(共 5 题, 50.0 分)

1.(其它, 10.0 分)

1.在书库中查找指定的图书。在书库文本文件中，查找指定的书名字符串，书库中每本书占据一行，包含书名、定价。从键盘输入字符串 s，代表要查找的书名。如果在书库中找到了该书，那么就把该行在屏幕上显示出来，并统计该书在书库中出现的次数。

2.(10.0 分)

2.设计一个类 DataArray,它具有一个 int 指针成员。构造函数具有一个整形参数 c，为指针成员分配 c 个 int 类型的数据空间，并采用随机函数初始化。析构函数释放指针指向的空间。另外，设计一个函数 getMax,返回这些数中的最大值。主函数不写。

```
class DataArray
{
public:
        DataArray(int c ); //编程实现 1
        ~DataArray( );     //编程实现 2
        int getMax( );     //编程实现 3
private:
        int   length; //  代表元素个数
        int *pint; //  指向 C 个整形元素的空间指针
};
```

3.(10.0 分)

3.一个类 PersonInfo 定义如下，具有一个 char *指针成员。要实现的成员函数包括对象构造、对象初始化，具体见 main 函数中的注释。在下面程序的基础上，完成该类的定义。

```
class PersonInfo
{
    char *name ;
    int   age;
public:
    ~PersonInfo( ){ delete [] name; }
    char * getName( ){ return name ; }
    int     getAge( ){ return age ; }
    //此处省略了构造函数、拷贝构造函数
};
void main( ){
   PersonInfo st1("Jim" , 20 )，st2("Bob", 18 ) ;   //功能 1
   PersonInfo st3 = st1 ;   //功能 2
}
```

4.(10.0 分)

4.豌豆射手参与了植物大战僵尸的战斗。

定义一个基类 Hero.其子类是 PeaShooter, 由于父类构造函数带参数，需要进行参数传递。在主函数中，定义一个子类对象。注意:需要编程实现的部分有 3 处。

```
class Hero
{
        char name[10] ;   //游戏角色名称
        int bloodAmount ; //游戏角色剩余血量
public :
        Hero(int bloodAmount, char *name); //构造函数初始化。需要编程实现 1
         int getBlood ( ){ return bloodAmount ; }
         void setBlood ( int x){ bloodAmount = x; }
 };
class PeaShooter : public Hero //  子类:豌豆射手
{
      int weapon; //  武器编号
public :
      PeaShooter(int bloodAmount，char *name, int weapon); //  子类构造函数。需要编程实现 2
      ~PeaShooter( ) ;
      void PlantsVSZombies(); //  战斗模拟。通过循环模拟战斗到最后一滴血，调用 getBlood ()
                  //和 setBlood ( int )，当血量为 0 时结束战斗。需要编程实现 3
};
void main( ){
      PeaShooter player1(100, "PS1",7); //  三个参数分别是血量、战士名称、武器编号
      player1.PlantsVSZombies(); //  参与战斗
}
```

5. (10.0 分)

5. 假设一个武器系统包含常规武器和战略武器。定义一个武器类 weapon (成员变量包括:普通成员: conventional, strategic, 其它必要成员函数自行补充)。要求:

(1)重载运算符“+”求两种武器数量之和(重载为类的友元函数)。

(2)重载运算符“>”比较武器总数量的多少(重载为类的成员函数)。

(3)编写主函数进行测试。

一、

1、

运行结果：

3

0

5

8

2、

运行结果：

D

d

d

D

3、

运行结果：

Constructor

Constructor

Younger

0

Deconstructor

Deconstructor

4、

运行结果：

6　2

5、

运行结果：

Letter:C

Letter:C

Letter:B

Letter:C

6、

运行结果：

Base

Derived

Copy Base

12

7

~Derived

~Base

~Base

7、

运行结果：

B

BFun()

~B

8、

运行结果：

Part:2

Part:5

Whole

9、

运行结果：

A2

C

9

~A

~C

10、

运行结果：

4

8

1

7

二、1

```
#include <stdio.h>
#include <string.h>
typedef struct book{
  char bName[80];
  float bPrice;
}date;

void input (FILE *fp)
{ fp = fopen("book.txt", "a");
  date d[10];
  for(int i=0;i<5;i++)
  {
    scanf("%s%f", d[i].bName, &d[i].bPrice);
  }
    fclose(fp);
}
void sreach(FILE *fp)
{ fp = fopen("book.txt", "r");
  int f=0;
  date t;
  char name[80];
  gets(name);
  for(int i=0;i<5;i++)
  {
    fscanf(fp,"%s%s%f",t.bName,t.bAuthor,&t.bPrice);
    if(strcmp(user,t.bName)==0){
      f++;
      if(f==1)
      printf("书名%s 价格%.1f",t.bName,t.bPrice);
    }
  }

  if(f==0){
    printf("查无此书");
  } else {
    printf("出现次数:%d",f);
  }
    fclose(fp);
}
```

二、2

```
DataArray::DataArray(int c)
{
  length=c;
  pint=(int*)malloc(c*sizeof(int));
  for(int i=0;i<c;i++)
    pint[i]=rand()%100;
}

DataArray::~DataArray()
{
  length=0;
  free(pint);
}

int::DataArray::getMax()
{
  int res=pint[0];
  for(int i=0;i<length;i++)
    if(res<pint[i])
      res=pint[i];
  return res;
}
```

二、3

```
PersonInfo(char *n,int a)
{
   strcpy(name,n);
   age=a;
}

PersonInfo(PersonInfo &p)
{
   strcpy(name,p.getName());
   age=p.getAge();
}
```

二、4

```
Hero::Hero(int bloodAmount,char *name)
{
  this->bloodAmount=bloodAmount;
  strcpy(this->name,name);
}

PeaShooter::PeaShooter(int bloodAmount,char *name) {
  Hero(bloodAmount,name);
  this->weapon=weapon;
}

void PeaShooter::PlantsVSZombies()
{
  while(getBoold()<=0)
    return;
}
```

二、5

```
class weapon
{
  int sum;
public:
  weapon(int s){sum=s;}
  friend weapon operator+(const weapon &c1,const weapon &c2);
  bool operator >( weapon&);
};


bool weapon::operator>( weapon& c)
{
  return c.sum < this->sum;
}

weapon operator+(const weapon & c1, const weapon & c2)
{
  weapon c3(0);
  c3.sum = c1.sum + c2.sum;
  return c3;
}
```