一、 单选题

1	2	3	4	5	6	7	8	9	10
D	C	A	A	A	D	B	D	B	D
11	12	13	14	15	16	17	18	19	20
C	D	C	B	B	B	D	A	D	A
21	22	23	24	25	26	27	28	29	30
A	C	C	C	C	D	C	A	D	A
31	32	33	34	35	36	37	38	39	40
D	C	B	A	A	B	B	D	C	C
41	42	43	44	45	46	47	48	49	50
A	C	B	B	A	D	A	C	C	C
51	52	53	54	55	56	57	58	59	60
C	C	C	D	B	B	A	B	C	D
61	62	63	64	65	66	67	68	69	70
A	C	B	B	D	B	B	D	C	C

二、 判断题

1	2	3	4	5	6	7	8	9	10
×	√	√	×	√	×	√	×	×	√
11	12	13	14	15	16	17	18	19	20
√	×	√	√	√	×	×	×	×	×
21	22	23	24	25	26	27	28	29	30
√	×	√	×	√	×	√	×	×	×
31	32	33	34	35	36	37	38	39	40
√	×	×	×	×	×	×	×	√	√
41	42								
×	√								

三、 程序分析

1. 正确。 因为有转型构造函数 Clock(int k)。

2. (1) 不会
   (2) `string s="abc"; b.A::f(s); int t=45; b.f(t)`

3. (1)调用了构造函数 `Triangle (int s1,int s2,int s3)`
   (2)调用了构造函数 `Triangle ( Triangle& )`或者拷贝构造函数
   (3)调用了构造函数 `Triangle ( )`

4. `a=10 b=20`
   `a=20 b=20`

5. set (const string &n) 中的 const 表明在该函数中不会修改形参 n 的值。
   get( ) const 中的 const 表明 get 是只读成员函数，不会在该函数中修改该类的任何数据成员的值。
   const string & get 中的 const 表明不会修改引用返回的变量的值。

6. 错误地方 CD( ){ }，原因：基类没有提供默认构造函数，派生类应显式调用基类的构造函数。

7. 15 11 12 13 14

8. 2
   -5
   0

9. 15 22
   7 33

10. ctor: x = 0
    ctor: x = 1
    copy ctor: x = 1
    dtor: x = 1
    dtor: x = 1
    dtor: x = 0

11. ctor: h = 0
    ctor: h = 0
    ctor: h = 20
    dtor: h = 20
    dtor: h = 20
    dtor: h = 0

12. 0: 0: 0
    0: 0: 0
    21: 58: 59
    21: 59: 0

13. **BC constructor**
    DC constructor
    DDC constructor
    DDC destructor
    DC destructor
    BC destructor

14. **100**
    200
    100
    100

-999
-999

15. **Isum=21, Dsum=26.9**

16. **C(int) firing**
10

17. 6 9
120 153

18. 2
-5
0

19. X=6,y=8
X=30,y=40

20. ctor: Hello
ctor: World
dtor: Hello
dtor: World

21. 5,10
2:8

22. 40 1 2 0 6 0 7 3 5

23. Global
Local
Local
Local
Local
Global

24. 0/12/7
9/7 9/7
9/7 16/7

25. n= 0
n= 1
n= 0
n= 1
n= 1
n= 0
n= 1

```
     n=  0
```

```
26.   Point  0
      Square  100
      Circle  314
```

```
27.   9    6    3    6
```

四、程序设计

参考代码如下：

1.

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
void main(){
    vector<int> v(10);
    int i;
    for(i=0;i<10;i++)
        cin>>v[i];
    sort(v.begin(),v.end());
    for(i=0;i<10;i++)
        cout<<v[i]<<endl;
}
```

2.

```
#include <iostream>
#include<string>
using namespace std;
class Student{
public:
    Student(string s1,string s2,int a ){
        name=s1;
        no=s2;
        age=a;
    }
    void print(){
        cout<<"name:"<<name<<endl
            <<"No:"<<no<<endl
            <<"age:"<<age<<endl;
    }
private:
    string name;
    string no;
    int age;
```

```
};
void main(){
    Student p("Tom","140001",18);
    p.print();
}
```

3.

```
#include <iostream>
using namespace std;
class Point{
public:
    Point(){x=0;y=0;}
    Point(int i,int j){x=i;y=j;}
    bool operator==(Point &t){ return x==t.x&&y==t.y;}
    friend istream & operator>>(istream&in,Point&t);
    friend ostream & operator<<(ostream&out,Point&t);
 private:
       int  x;    int  y;
 };
istream & operator>>(istream&in,Point&t)    {   return in>>t.x>>t.y;  }
ostream & operator<<(ostream&out,Point&t)   {   return out<<"x="<<t.x<<"y="<<t.y<<endl;}
void main(){
    Point a,b;
    cin>>a; cin>>b;
    if(a==b)
        cout<<a;
    else
        cout<<b;
}
```

4.

```
#include <iostream>
using namespace std;
 class  Shape{
 public: virtual void GetGetArea()=0;
 };
class  Rectangle:public Shape{
 private:   int x  ;  int  y;
 public:  Rectangel(int i, int j){x=i; y=j;}
  void GetGetArea(){cout<<" Rectangel   Area:"<<x*y ;}
};
class  Circle : public Shape{
 private:  float   r;
 public :Circle (float  i ){ r=i; }
  void GetGetArea(){cout<<" Circle   Area:"<<3.14*r*r<<endl;}
```

```
};
void main( ){
  Shape *p[2];
 Rectangle   rec(7,9);   Circle  cir( 9);
 p[0]=& rec; p [0]-> GetGetArea();
 p[1]=& cir; p [1]-> GetGetArea();
 }
```

5.

```
#include<iostream>
   using namespace std;
   class  Complex{
       double  real, imag;
   public:
      Complex(double r=0,double i=0) { real = r; imag = i;}
       Complex operator+( Complex&);
       friend bool operator ==( Complex &, Complex &);
       friend istream & operator>>(istream & input, Complex &c);
       friend ostream & operator<<(ostream & output, Complex &c);
 };
 Complex Complex::operator+( Complex& c){
   Complex c1(real+c.real, imag +c.imag);
    return c1;
}
bool  operator==(Complex& c,Complex& c1) {
  if(c1.real==c.real&&c1.imag==c.imag)
   return true;
 else
   return false;
}
ostream & operator<<(ostream & output,Complex &c){
     output<<c.real;
     if(c.imag>0)  output<<"+";
     if (c.imag!=0)  output<<c.imag<<"i";
     output<<endl;
     return output;
}
istream & operator>>(istream & input,Complex &c){
     input>>c.real>>c.imag;
     return input;
}
```

6.

```
#include<iostream>
using namespace std;
#define  PI  3.1416
```

```
// Model 类声明
class  Model {
public:
   Model(double r)  { m=r; }
   virtual   double volume() const=0;
protected:
   double  m; //1 分
}
//  Cube:类声明与实现
class Cube: public Model {
public :
   Cube (double=0.0) ;
   virtual double volume() const;
};
Cube :: Cube (double a):Model(a){  }
double Cube:: volume () const    {
   return   m * m* m;
}
//  Cylinder类声明与实现
class Cylinder: public Model   {
 public :
    Cylinder (double=0.0, double=0.0) ;
    virtual  double volume() const;
 private:
     double   h;
};
Cylinder:: Cylinder (double a, double b):Model(a),h(b) {    }
double Cylinder:: volume () const    {
 return  PI* m * m* h ;
}
//测试程序
int main()
 {
 Cube   cub(2.4) ; //1 分
 Cylinder cyl(1,2); //1 分
 Model *p[]={&cub, &cyl};//1 分
 for(int  i=0;i<2;i++)  //2 分
      cout<<" volume ="<<p[i]->volume()<<endl;
  return 0;
 }
```

7.

```
class Student  {
public：
string name;  //姓名
int    id;    //学号
```

```
double score; //成绩
Student (const char*  str,  int i,  doulole s) :name (str) ,  id (i) ,  score (s) {}
};
template <typename T, class Function>
void rnysort (T first,  T last,  Function f)  {
       T p (last) ;  advance (p,  -1) ;
     for (T i  = first; i != p; ++i  {
              T j (i) ;  advance (j,  1) ;
           for ( ; j != last; ++j  {
                  if  (f(*i,  *j)  {
                   typename  iterator  traits<T> : : value_type  t(*i)
                            *i = *j, *j = t;
                     }
                }
         }
}
bool cmp_by_score (const Student& a,  const Student& b){
     return a.score < b.score;
}
void print (const Student&  s)  {
      cout<<s.name<< "\t"<< s.id<<"\t"<< s.score<< endl;
}
int main()  {
    Student  s[] = {Student(’ ’ 1’ ’ , 1, 80),  Student( “2”,  2,  7 0),Student( “3”,  3,
90),  Student(“4”,  4,  9 7)} ;
    int  n  =  sizeof (s)  /  sizeof (*s);
    mysort (s,  s  +  n,  cmp_by_score);
    for_each (s,  s  +  n,  print);
)
8.
class Point {
private :
      double x. y, z;
public :
      Point (doulole  a,  doulole b,  doulole  c) :x (a) ,  y(lo) ,  z (c) { }
      double operator - (const Point& rhs) {
            doulole  dx = x - rhs.x;
            double  dy = y - rhs .y;
            double  dz = z - rhs .  z;
          return sqrt (dx *dx + dy *dy + dz *dz);
        }
};
class Sphere {
private :
      Point  center;
      doulole radius;
```

```
public :
     Sphere (const Point& p,  double r): center (p) ,  radius (r) { }
      double operator - (const Sphere& s) {
          return  center - c. center - radius - s . radius;
};
int  main ( )   {
      Point  pe (0,  0,  0) ,  pm (100,  100,  100) ;
      double re  =  0, rm= 5;
     Sphere  Earth (pe,  re) ,  Moon (pm,  rm) ;
      double d = Earth - Moon;
       cout<< d<<endl;
}
```

9.

```
class ComputerAccessory {
private :
string manufacturer; //制造商
doulole price;    //价格
public :
     ComputerAccessory (const  char*  mf  =  ″ ″,   doulole  p=0) :manu fa cture r (mf ) ,
price (p){}
      doulole  GetPrice ( )  const   { return  price; }
};
class MotherBoard : public ComputerAccessory {
private :
     string chipset;  //"Intel",  “AMD"
public :
MotherBoard ( const   char* mf= “” ,   doulole  p = 0,   const  char* c=” ”)
     : ComputerAccessory (mf, p) , chipset ( c)
     {}
};
class Memory : public ComputerAccessory {
private :
     int capacity;    //IG, 2G
public :
     Memory  const  char*  mf  = ″ ″,   doulole p = 0, int  c =0)
       : CornputerAccessory (mf,   p) , capacity ( c)
     {}
class Monitor : public ComputerAccessory {
private :
     string mtype; // "CRT", "LCD"
public :
      Monitor (const char* mf = “” , double p = 0, const char* m= ″ ″ ) : ComputerAccessory
(mf,  p) ,  mtype (m) {}
class Computer {
private :
```

```
    MotherBoard    mboard;
    Memory mem;
    Monitor mtype;
public :
    Computer (const MotherBoard& mb,  const Memory& me,  constMonitor& mt) : mboard (mb ) ,
mem (me ) , mtype (mt )   {}
     double  TotalPrice ( )   {
         return  mboard. GetPrice ( ) +  mem. GetPrice ( ) +mtype . GetPrice ( ) ;
     }
};
int  main ( )   {
   MotherBoard  mb( "Intel",1000,"Intel");
   Memory  me( "Kingmax",1000,  2);
   Monitor  mt(" Founder",1000,  ' ' LCD' ' );
   Computer  c(mb,  me,  mt);
   cout << c.TotalPrice() <<  endl;
)
```

10.(略)
11.(略)
12.(略)
13.(略)
14.(略)
15.(略)