C++ 第二次阶段性测试（满分 60）

一 、单项选择题（16 分, 每题 2 分）

1. Which of the following is correct about the overloading mechanism in C++? (　　).

[A] You can overload normal functions, and you can also overload member functions of classes

[B] Operators can only be overloaded as member functions

[C] Constructors and destructors can be overloaded

[D] Function templates can not be overloaded

2.　There are three classes as follows. If object **D d** is created, the output is（ ）

```
class A
{   public:   A(){cout<<”A”;}
};
class B
{   public:   B(){cout<<”B”;}
};
class D：public B
{     A a ;
  public:
      D( ){cout<<”D”}
};
```

[A]　ABD　　[B]　BAD　　[C]　DBA　　[D]　DAB

3. Which of the following is **incorrect** about polymorphism in C++? (　)

[A]　Polymorphism means that when different objects receive the same message, they perform different behaviors

[B] Polymorphism is divided into compile-time polymorphism and run-time polymorphism

[C]　Compile-time polymorphism is also called static polymorphism, which can be achieved by function overloading

[D]　Run-time polymorphism is also called dynamic polymorphism, which can be achieved by templates

4. According to the following function template, which of the following is correct?

```
template<class T>
T add(T& a, T b){ return a+b; }
```

[A]　int x,y; add(x, y);　　　　　　[B]　int x; float y; add(x,y);

[C]　double x; float y; add(x,y);　　　[D]　double x; int y; add(x,y);

5. Considering the base class X and its derived class Y, which one is correct?（　　）

【A】Class Y cannot declare same-name data as class X.

【B】X x;　Y y;　X* px =&y ; Y* py = &x;

【C】X x;　Y y;　X& rx = y ; Y& ry = x;

【D】X x;　Y y;　X　xx = y ;　xx= x;

6. About the following two classes, which of the following is **incorrect**?（ ）

```
class Plant
{
public:
  Plant ( char*) ;
  virtual void water( ) ;
  virtual void sprout ( ) =0;
protected:
      char name[10];     };
```

```
class Tree: public Plant
{
public:
       Tree ( char* );
       void water( );
       void fertilize ( );
};
```

[A] You can define Plant * p or Tree * p.
[B] Plant is an abstract class, so you can't create objects of this class
[C] void function(Tree & t);   //this function is legal(合法)
[D] Plant function(Plant * p); // this function is legal(合法)

7. About the following three classes, which of the following is **incorrect**?（ ）

```
class A{
protected:   int x;
   public :      void   set_X( int i){ x = i} ;
                 int   get_X( ){ return x; } ;
};
class B : protected A
{      int y;   };
class C: private B{…};
void main( ){
   B objB;     …
}
```

[A] The x in class A can be accessed directly in class C.
[B] objB in the main function can call the set_X(int i) function in class A.
[C] get_X() can be accessed directly in class C.
[D] You can't access y of class B in class C.

8. About “exception handling mechanism” in C++, which one is correct? （　　）
[A] If there is only one statement in try block, { } of try block can be omitted(删除).
[B] If there is only one statement in catch block, { } of catch block can be omitted.
[C] Exception thrown may(可能) cause automatic(自动的) execution of destructor.
[D] The catch block after the try block must be executed.

二、写出下面程序的运行结果。（17 分）

1. （6 points）

```
#include<iostream>
#include<string>
using namespace std;
class Animal
{
protected:
        string name ;
public:
        Animal(string n):name(n)
        {       cout<< "Animal 1"<<endl;     }
        Animal():name("无名")
        {       cout<< "Animal 0"<<endl;     }
        void show()
        { cout<<name<<endl;    }
};

class Cat: public Animal
{
        int year;
public:
        Cat( )
        {            cout<<"Cat 0"<<endl; year= -1;       }
        Cat(string n, int y):Animal(n),year(y)
        {            cout<<"Cat 1"<<endl;       }
            void show()
        { cout<<name<<" "<<year <<" years old\n";        }
};

int main(){
      Cat cat1, cat2("Garfield",3);
      cat1.show();
      cat2.show();
}
```

2. （5 points）

```
#include<iostream>
#include<string>
using namespace std;
class Animal
{
protected:
        string name ;
public:
        Animal(string n = "无名"):name(n){ }
        virtual void shout()
        { cout<< name<< endl; }
};

class Cat: public Animal
{
        int year;
public:
        Cat( ) :Animal("无名"), year(0){ }
        Cat(string n, int y):Animal(n)
```

```
    {       year = y>0 ? y:0;       }
    void shout();
};
void Cat::shout()
{
    string result = name;
    result += ":miao miao miao";
    cout<<   result << endl;
}

class Dog: public Animal
{
    int shoutNum;
  public:
    Dog(string n, int y):Animal(n)
    {        shoutNum = y>0 ? y:0;    }
    void shout();
};
void Dog::shout()
{
    string result = name;
    for(int i = 0; i < shoutNum; i++)
        result += ":汪";
    cout<<   result << endl;
}

void main()
{
    Animal a;
    Cat c("Garfield",3);
    Dog d( "阿福",2);

    Animal* pA[3] ={&a, &c,&d};
    for(int i=0; i<3; i++)
      pA[i]->shout();

    a = c;
    a.shout();
    a = d;
    a.shout();
}
```

3.   (6 points)

```
#include <iostream>
using namespace std;
template <typename T>
T Max(T x,T y) {
    cout<<"Max_1"<<" ";
    return x > y ? x : y ;
  }
template <typename T>
T Max(T x,T y,T z){
  cout<<"Max_2"<<" ";
    T temp = Max(x,y) ;
    return Max(temp,z) ;
}
double Max(double x,double y) {
   cout<<"Max_3"<<" ";
   return x > y ? x : y ;
}
void main ( ) {
    cout<<Max(1.0, 2 )<<endl ;
    cout<<Max(2.0, 5.0)<<endl ;
    cout<<Max('a', 'z', 'b')<<endl ;
}
```

三 填空题（每空一分，共 9 分，每行不能填写超过一个语句）

```
#include <iostream.h>
#include <process.h>
const int MAX = 10;                    // max capacity of Stack
template<typename T>
class Stack
{
    private:
        T st[ _(1)________ ];
        int top;                       // top of the stack
    public:
        Stack() { top = -1; }
        void push( const T& number) ;
        T pop();
 };
（2）________________________
（3）______________________ //define push( )
{
            if(__（4）__________________ )   // stack is full
            {
                cerr<<"Stack is Full!"<<endl;
                abort();
            }
            __（5）_________________
 }
__（6）___________________
__（7）___________________// define pop ( )
{
            if( __（8）___________)          // stack is empty
            {
                cerr<<"Stack is Empty!"<<endl;
                abort();
            }
            return st[ _(9)__________ ];
}
```

四、编程题（8 分）

Given class Room:

```
#include < iostream >
using namespace std;
class Room
{
     protected:
        double length;      // length of Room
        double width;      // width of Room
     public:
        Room(double l, double w) :length(l),width(w){}     //constructor
};
```

Please design and define class ClassRoom which is a derived class of Room. ClassRoom has a *seat* property which represents the number of seats in the classroom. ClassRoom also has function *calAverArea* which returns average area per seat. In main function, an instance of ClassRoom is created and *show* function is called.

```
void main(){
     ClassRoom cr(30, 10, 12);      // 30 seats, length is 10, width is 12
     cr.show();                     //show on the screen      length:10, width:12, seat:30, averarea:4
}
```

Please design and define class ClassRoom:

四、程序设计题(10 分)

Design a program framework for book management. ***Reader*** can borrow different types of ***Publication***. ***Publication*** can be divided into ***Book*** and ***Magazine***. ***Publication*** has ***name*** and ***id***. In addition, ***Book*** should have the ***author*** information and ***Magazine*** needs to save its ***volume*** and period ***number***. All the information of a ***Publication*** can be print out by ***print***(), but ***Book*** and ***Magazine*** have different information to print. ***Reader***'s attributes include ***name, array of Publications*** and current ***number*** of publications he (or she) has borrowed. Each reader can borrow up to 10 publications. Readers can display information of all publications by ***show()***.

(1) Please design all the classes mentioned. Declare attributes and functions (only needed). Class implementation is not required. Pay attention to the class relations and virtual functions.

(2) Please write the definition of ***show()*** in ***Reader***. In this function, the print() functions of all the publications borrowed by reader are called and output their different information.