一．选择题（每小题 2 分，共 20 分）

1．下面对于类的描述，是正确的是（　　）

A．类是抽象数据类型的实现

B．类是具有共同行为和属性的若干对象的统一描述体

C．所有的类都能创建对象

D．类就是 C 语言中的结构类型

2．下列关于 C++函数的说明中，正确的是 (　　)

A．内联函数就是定义在另一个函数体内部的函数

B．函数体的最后一条语句必须是 return 语句

C．标准 C++要求在调用一个函数之前，必须先声明其原型

D．编译器会根据函数的返回值类型和参数表来区分函数的不同重载形式

3．下列不是描述类的成员函数的是(　　)

A．构造函数　　B．析构函数

C．友元函数　　D．拷贝构造函数

4．构造函数不具备的特征的是(　　)

A．构造函数的函数名与类名相同　　B．构造函数可以重载

C．构造函数可以设置默认参数　　D．构造函数必须指定类型说明

5．下面有关重载函数的说法中正确的是（　　）

A．重载函数必须具有不同的返回值类型；

B．重载函数形参个数必须不同；

C．重载函数必须有不同的形参列表

D．重载函数名可以不同；

6．下面关于 C++中类的继承与派生的说法错误的是（　　）

A．基类的 protected 成员在公有派生类的成员函数中可以直接使用

B．基类的 protected 成员在私有派生类的成员函数中可以直接使用

C．私有派生时，基类的所有成员访问权限在派生类中保持不变

D．继承可以分为单一继承与多重继承

7．下面关于运算符重载的描述错误的是（）

A．运算符重载不能改变操作数的个数、运算符的优先级、运算符的结合性和运算符的语法结构

B．不是所有的运算符都可以进行重载

C．运算符函数的调用必须使用关键字 operator

D．在 C++语言中不可通过运算符重载创造出新的运算符

8．关于虚函数的描述中，（ ）是正确的。

A．虚函数是一个 static 类型的成员函数

B．虚函数是一个非成员函数
C．.基类中说明了虚函数后，派生类中将其对应的函数可不必说明为虚函数
D．派生类的虚函数与基类的虚函数具有不同的参数个数和类型

9．假定 AB 为一个类，则执行 AB x；语句时将自动调用该类的(　　)
A．有参构造函数　　B．无参构造函数
C．拷贝构造函数　　D．赋值构造函数

10．下面关于编写异常处理代码规则中不正确的是(　)
A. 可以有数量不限的 catch 处理程序出现在 try 块之后，在 try 块出现之前不能出现 catch 块。
B. try 块中必须包含 throw 语句。
C. 在关键字 catch 之后的圆括号内的数据声明必须包括其类型声明。
D．如果 catch 中处理程序执行完毕，而无返回或终止指令将跳过后面的 catch 块继续执行

二．改错题(共 24 分)

1．指出下面程序段中的错误，并说明出错原因（14 分）

```
class X{
private:
    int a=0;                    //A 行
    int &b;                     //B 行
    const int c;                //C 行
    void setA(int i){a=i;}      //D 行
    X(int i){a=i;}              //E 行
public:
    int X():b(a),c(a){a =0;}                //F 行
    X(int i,int j,int k):b(j),c(k){a=i; }   //G 行
    static void setB(int k){b=k;}           //H 行
    setC(int k)const {c=c+k;}               //I 行
};
void main()
{
    X x1;                       //J 行
    X x2(3);                    //K 行
    X x3(1,2,3);                //L 行
    x1.setA(3);                 //M 行
}
```

2. 指出下面程序段中的错误，并说明出错原因（6 分）

```
#include<iostream>
using namespace std;
class Base1 {
    int b1;                                         //A 行
```

```
public:
    Base1(int b1=0) {this->b1=b1;}                          //B 行
    void f(){ cout<<"From    Base1"<<endl;}                 //C 行
};
class Base2 {
    int b2;                                                 //D 行
public:
    Base2(int b2){this->b2=b2;}                             //E 行
    void f() { cout<<"From    Base2"<<endl;}                //F 行
};
class Derived: public Base1, public Base2 {                 //G 行
    int d;
public:
    Derived(int d){this->d=d;}                              //H 行
    void g(){ cout<<"From    Derived"<<b1<<b2<<endl; }      //I 行
};
void main(){
    Derived dObj(10);                                       //J 行
    dObj.f();                                               //K 行
    dObj.Base1::f();                                        //L 行
}
```

3. 指出下面程序段中的错误，并说明出错原因（4 分）

```
template<class TYPE>
TYPE max(TYPE a,TYPE b)
{
    return a>b?a:b;
}
main()
{
    max(2,3);              //A 行
    max(3.0,3.14);         //B 行
    max(3,3.14);           //C 行
}
```

三．阅读程序，写出程序的运行结果（共 32 分）

1.（4 分）

```
#include<iostream>
using namespace std;
class Implementation{
public:
    Implementation(int y){value=y;}
    void setValue(int v){value=v;}
```

```
    int getValue() const {return value;}
private:
    int value;
};
class Interface{
public:
    Interface(int);
    void setValue(int);
    int getValue() const;
private:
    Implementation *ptr;
};
Interface::Interface(int v):ptr(new Implementation(v)){}
void Interface::setValue(int v){ptr->setValue(v);}
int Interface::getValue() const {return ptr->getValue();}
void main()
{
    Interface i(5);
    cout<<i.getValue()<<endl;
    i.setValue(10);
    cout<<i.getValue()<<endl;
}
```

2.（12 分）

```
#include<iostream>
using namespace std;
class B1
{
public:
    B1(int a){cout<<"constructing B1 "<<a<<endl;}
};

class B2:public B1{
public:
    B2(int b,int a):B1(a){cout<<"constructing B2 "<<b<<endl;}
};

class B3:public B2
{
public:
    B3(int a,int b,int c,int d,int e):B2(a,b),memberB2(c,d),memberB1(e)
    {cout<<"constructing B3"<<endl;}
private:
    B1 memberB1;
```

```
    B2 memberB2;
};

void main()
{
   B3 b3(1,2,3,4,5);
}
```

3.（8 分）

```
#include<iostream.h>
class Character
{
    char i;
public:
    Character (char a=0){i =a; }
    Character operator ++();
    Character operator ++(int);
    void print(){cout<<i<<endl;}
};
Character Character::operator ++()
{
    i++;
    return*this;
}
Character Character::operator ++(int)
{
    Character j;
    j.i=i++;
    return j;
}
void main()
{
    Character    x(65), y(98), z;
    z = ++x;
    x.print();
    z.print();
    z = y++;
    y.print();
    z.print();
}
```

4.（8 分）

```
#include <iostream>
```

```
using namespace std;
enum errs{error0,error1};
double Divide(double test1, double test2)
{
    try{
        if(test2==0) throw error0;
        if(test2>=1000) throw error1;
    }
    catch(errs er){
        switch(er)
        {
            case error0:
                cout<<"除数不能为 0!"<<endl;
                break;
            case error1:
                cout<<"除数太大!"<<endl;
                break;
            }
        }
    return test1/test2;
}
void main()
{
    cout<<Divide(2,0)<<endl;
    cout<<Divide(1,1000)<<endl;
}
```

四、问答题（10 分）

```
#include <iostream.h>
class A{
private:
  //…其它成员
public:
  virtual void func(int data){cout<<”class A:”<<data<<endl;}
  void func(char *str){ cout<<”class A:”<<str<<endl; }
};

class B: public A{
//…其它成员
public:
  void func() {cout<<”function in B without parameter! \n”;}
  void func(int data) { cout<<”class B:”<<data<<endl; }
  void func(char *str){ cout<<”class B:”<<str<<endl;}
```

```
};
int main()
{
   A *pA;
   B b;
   pA=&b;
   pA->func(1);
   pA->func("haha");
   return 0;
}
/*************************************************************************/
```

问题 1：（本小题 4 分）写出程序的运行结果：

问题 2：（本小题 2 分）若想在函数 main()中通过 pA 调用类 B 中定义的参数表为空的函数 func() ，如 pA->func();是否正确？

问题 3：（本小题 4 分）如果要记录已经创建的 A 类的实例（对象）的个数，我们可以借助于类的静态成员。修改上面类 A 的定义，使得它包含一个私有的静态成员 object_count，记录属于该类的对象的个数，然后为类 A 增加必要的成员函数，使得下面的程序：

```
void main()
{
   A *pA=new A[3];
   cout<<"There are "<<pA->GetObjectCount()<<" objects"<<endl;
   delete []pA;
   cout<<"There are "<<A::GetObjectCount()<<" objects"<<endl;
}
```

得到的输出为：

```
There are 3 objects
There are 0 objects
```

请写出类 A 的定义（将所有的函数成员实现写在类定义体中）和初始化类的静态成员 object_count 的语句。

五．程序设计题（14 分）

1. （4 分）修改下面给出的程序，但不允许对 main()函数作任何修改，使程序能够在屏幕上输出唐诗：

白日依山尽，
黄河入海流。
欲穷千里目，
更上一层楼。

原来的程序为：

```
#include<iostream.h>
void main()
{
cout<<"欲穷千里目，"<<endl;
}
```

2. （10 分）定义一个大学生类 student，函数私有数据成员：姓名、学号、校名，并为它定义带参数的构造函数、参数带缺省值的构造函数和输出数据成员值的 print()公有成员函数，另定义研究生类，它以公有继承方式派生于类 student，新增加“研究方向、导师名”两个私有数据成员，并定义带参数的构造函数和输出研究生数据的 print()公有成员函数。在 main()函数中定义基类和派生类对象，对类进行测试。

主函数的测试程序如下：

```
void main()
{
    Student stu1("Li","1600141","XingJiang University");
    stu1.print();
    GraStudent gstu("Wang","1600240","XJUniversity","Computer","Zhang");
    gstu.print();
}
```

程序运行输出结果如下：

```
name=Li
StuNum=1600141
universty_name=XJU

name=Wang
StuNum=1600240
universty_name=XJU
special is Compute
director is Zhang
```