

2-1 分数 4

下列程序代码，正确的输出结果是（）

```
1 int a=0xfe;
2 char b=062;
3 cout<<a<<" "<<b<<endl;
```

☐ A. 0xfe,062

☐ B. fe,62

☒ C. 254,2

☐ D. 254,62

2-2 分数 4

下列代码能正确执行的是（）

☒ A.

```
1 char a='C';
2 const char& p=a;
```

☐ B.

```
1 char a='C';
2 char* const p;
3 p=&a;
```

☐ C.

```
1 char a='C';
2 const char& p;
3 p=a;
```

☐ D.

```
1 char a='C';
2 char& p;
3 p=a;
```

2-3 分数 4

new和delete运算符正确的使用是（）

☐ A.

```
1 int* p=new int[10];
2 delete p;
```

☐ B.

```
1 int* p=new int(10);
2 delete []p;
```

☒ C.

```
1 vector<int>* v=new vector<int>[10];
2 delete []v;
```

☐ D.

```
1 int* a=new int[]{1,2,3,4};
2 delete []a;
```

2-4 分数 4

下列哪个代码能够实现x和y值的交换（ ）

☐ A.

```

1 void fun(int a,int b) {
2     int x = a;
3     a = b;
4     b = x;
5 }
6 int main() {
7     int x = 1, y = 2;
8     fun(&x, &y);
9     cout << x << ", "<<y << endl;
10    return 0;
11 }
```

☐ C.

```

1 void fun(int& a,int& b) {
2     int x = a;
3     a = b;
4     b = x;
5 }
6 int main() {
7     int x = 1, y = 2;
8     fun(&x, &y);
9     cout << x << ", "<<y << endl;
10    return 0;
11 }
```

☒ B.

```

1 void fun(int* a,int* b) {
2     int x = *a;
3     *a = *b;
4     *b = x;
5 }
6 int main() {
7     int x = 1, y = 2;
8     fun(&x,&y);
9     cout << x << ", "<<y << endl;
10    return 0;
11 }
```

☐ D.

```

1 void fun(const int&a,const int&b) {
2     int x = a;
3     a = b;
4     b = x;
5 }
6 int main() {
7     int x = 1, y = 2;
8     fun(x, y);
9     cout << x << ", "<<y << endl;
10    return 0;
11 }
```

2-5 分数 4

下面有一个结构体，对结构体的错误使用是（ ）

```

1 struct Book{
2     string name;
3     double price;
4 }
```

☐ A. `1 Book b{"C++",20};`

☒ B. `1 Book b;
2 b->name="C++";
3 b->price=20;`

☐ C. `Book *b=new Book[2];
b[0].name="C++";
b[0].price=20;`

☐ D. `Book *p=new Book();
p->name="C++";
p->price=20;`

2-6 分数 4

下列种类的函数中，哪一种不是类的成员函数？

- ☐ A. 构造函数
- ☐ B. 析构函数
- ☒ C. 友元函数
- ☐ D. 拷贝构造函数

2-7 分数 4

父类Base和子类Derive的定义如下。请问在子类中，继承的父类成员f, x, y的访问控制权限分别是：

```
class Base
{
    public:
        void f();
    protected:
        int x;
    private:
        int y;
};

class Derive : protected Base
{
};
```

- ☐ A. public, protected, private
- ☐ B. public, public, public
- ☐ C. private, private, private
- ☒ D. protected, protected, private

当变量 x 的输入值是9时，下面程序的运行结果是：

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    int x;
    cin>>x;
    try
    {
        cout<<"begin"<<endl;
        if(x>100)
        {
            cout<<"1"<<endl;
            throw string("too big.");
        }
        else
        {
            cout<<"x="<<x<<endl;
        }
        cout<<"2"<<endl;
    }
    catch(string e)
    {
        cout<<"3:"<<e<<endl;
    }
    cout<<"end"<<endl;
    return 0;
}
```

☐ A. begin

1

2

end

☐ B. begin

1

3: too big.

2

end

☒ C. begin

x=9

2

end

☐ D. begin

1

3: too big.

end

2-9 分数 4

下列哪个代码不会调用对象的拷贝构造函数 ()

☐ A.

```
1 MyClass a;  
2 MyClass b=a;
```

☐ B.

```
1 MyClass a;  
2 MyClass b(a);
```

☒ C.

```
1 MyClass a;  
2 MyClass& b=a;
```

☐ D.

```
1 void f(MyClass obj)  
2 {  
3     ....  
4 }  
5 MyClass a;  
6 f(a);
```

2-10 分数 4

根据下列类模板声明，正确初始化对象的方式是（ ）

```
1  template<typename T1, typename T2>
2  class MyClass{
3      private:
4          T1 x;
5          T2 y;
6      public:
7          MyClass(T1 _x, T2 _y):x(_x),y(_y){}
8  };
```

☒ A. 1 MyClass<int,char> a(10,'a');

☐ B. 1 MyClass a(10,'a');

☐ C. 1 MyClass<int,char> a;

☐ D. 1 MyClass a;