21-22 期末试卷参考答案（仅供参考，概不负责！）

1. 课本 P52 P53
2. 课本 P68
3. 课本 P343
4. 课本 P217
5.

需要添加一个花括号{}

```
if (expenseType == TYPE_DEVICE)//设备费
{
    if (price >= 10000 )
        cout << "需要招投标";
}
else if (expenseType == TYPE_ CONFERENCE )//会议费
    cout << "需要会以通知";
else
    cout << "一般报销流程";
```

6.

错误：继承类无法直接访问基类的私有成员

改正：可以将 Window 类里面的 height 和 width 改为 public 访问类型

或者在 MyWindow 的 private 里加 height 和 width 两个成员变量

7.

```
Run function funOne for var *p      :  1  2  1
Run function funOne for var r       :  1  2  1
Run function funTwo for var a          :  1  2  2
Run function funTwo for var *p      :  2  3  3
Run function funThree for var a     :  3  4  4
Run function funThree for var r     :  4  5  5
```

8.

```
A::say
A::sayVirtual
B::say
B::sayVirtual
A::say
B::sayVirtual
```

9.

```
#include<iostream>
using namespace std;
```

```
void Proportion(int score[],float &t9,float &t8,float &t7,float &t6,float &t5)
{
   for(int i=0;i<10;i++)
   {
      if(90<=score[i]&&score[i]<=100)
         t9+=0.1;
      else if(80<=score[i]&&score[i]<=89)
         t8+=0.1;
      else if(70<=score[i]&&score[i]<=79)
         t7+=0.1;
      else if(60<=score[i]&&score[i]<=69)
         t6+=0.1;
      else
         t5+=0.1;
   }
}
int main()
{
   int score[10]={78,100,65,49,92,72,85,99,88,91};
   float t9,t8,t7,t6,t5;
   Proportion(score, t9, t8, t7, t6, t5);
   cout<<"t9:"<<t9<<endl;
   cout<<"t8:"<<t8<<endl;
   cout<<"t7:"<<t7<<endl;
   cout<<"t6:"<<t6<<endl;
   cout<<"t5:"<<t5<<endl;

   return 0;
}
```

10.

```
#include<iostream>
using namespace std;
class Complex
{
public:
   Complex(float real=0.0,float imag=0.0):real(real),imag(imag){};
   Complex operator +(const Complex &c2);
   Complex operator -(const Complex &c2);
   Complex operator =(float a);
   Complex &operator --();
   friend ostream & operator<<(ostream &out,const Complex &c);
```

```
private:
    float real;
    float imag;
};
Complex Complex::operator=(float a)
{
    return Complex(a,0);
}
Complex Complex::operator+(const Complex &c2)
{
    return Complex(real+c2.real,imag+c2.imag);
}
Complex Complex::operator-(const Complex &c2)
{
    return Complex(real-c2.real,imag-c2.imag);
}
Complex &Complex::operator--()
{
    --real;
    --imag;
    return *this;
}
ostream & operator<<(ostream &out,const Complex &c)
{
    out<<"("<<c.real<<","<<c.imag<<")";
    return out;
}

int main()
{
    Complex d1;
    Complex d2(5.4,-2.5);
    Complex d3=20;
    cout<<"d1+d2="<<d1+d2<<endl;
    cout<<"d2-d3"<<d2-d3<<endl;
    cout<<"--d3="<<--d3<<endl;
    return 0;
}
```