

4.1-4.6

1. public和private是对类成员访问权限的控制，public可以被任意函数访问，private只能由本类的成员函数访问。

2. 保护类型成员的性质和私有成员性质相似，差别在于继承过程中对产生新类影响不同。

3. 构造函数在声明类对象时调用，用于初始化对象内成员，析构函数在对象生存期结束时调用，完成删除对象前的清理工作，比如delete掉使用new来申请的内存

4. 可以

5. 可以，a1 a2内的数据没有关系。

6. 复制构造函数是用一个已经存在的对象去初始化一个同类的新对象，在直接调用构造函数形参为已经声明过的对象时和声明新对象时使用赋值运算符时，以及其他函数形参为类对象时（以值形式传递会进行复制操作）和返回值为类对象时候

```
1  class ca
2  {
3      //...
4  };
5  ca a(...);
6  ca b(a);
7  ca c=a;
8  fun(a);
9  fun2()
10 {
11     ca d;
12     //...
13     return d;
14 }
```

4.9

```
1  #include <iostream>
2  using namespace std;
3
4  struct Point
5  {
6      int x;
7      int y;
8  };
9
10 class Rectangle
11 {
12     public:
13         Rectangle(Point p1, Point p2)
14         {
15             this->p1 = p1;
16             this->p2 = p2;
```

```

17     }
18     int getArea()
19     {
20         return (p2.x - p1.x) * (p2.y - p1.y);
21     }
22     private:
23         Point p1;
24         Point p2;
25 };
26
27 int main()
28 {
29     Rectangle r1({0,0}, {2,2});
30     cout<<r1.getArea()<<endl;
31     return 0;
32 }

```

4.13-4.14

```

1  //4.13
2  #include <iostream>
3  using namespace std;
4
5  class Circle
6  {
7  public:
8      Circle(double r);
9      double getArea();
10
11  private:
12      double radius;
13  };
14
15  Circle::Circle(double r)
16  {
17      radius = r;
18  }
19
20  double Circle::getArea()
21  {
22      return 3.14 * radius * radius;
23  }
24
25  int main()
26  {
27      Circle c(10);
28      cout << c.getArea() << endl;
29      return 0;
30  }

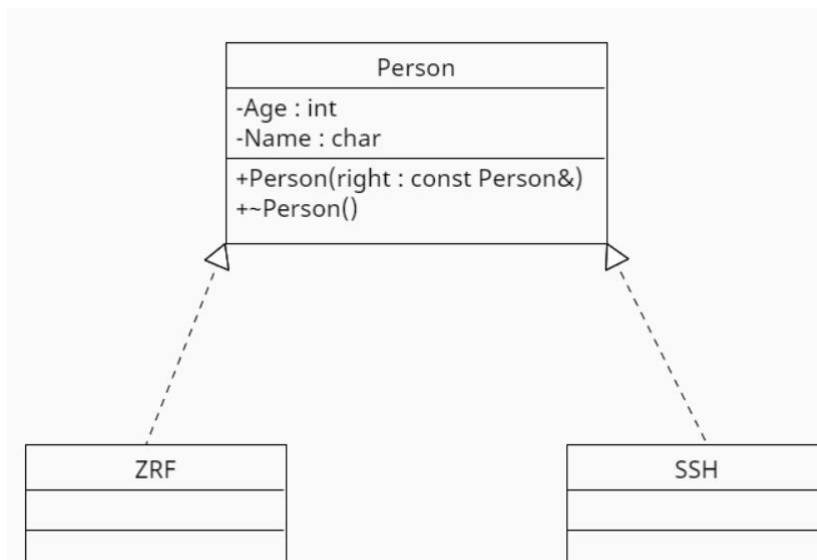
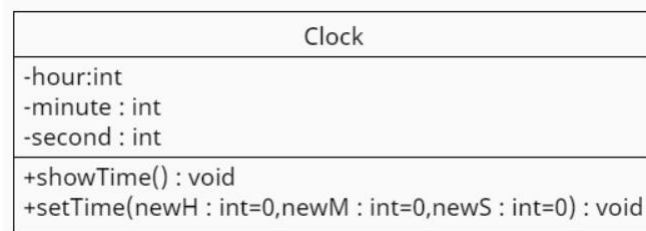
```

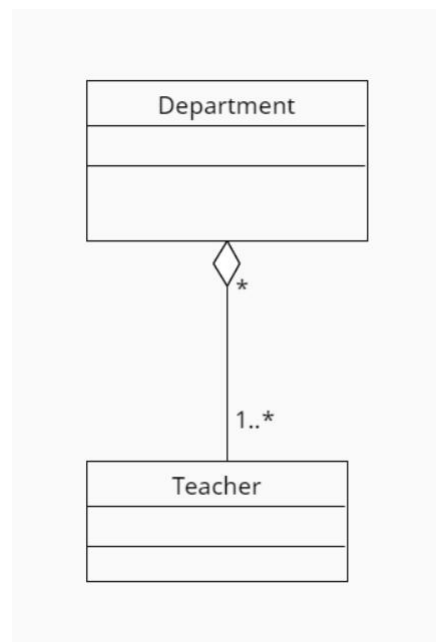
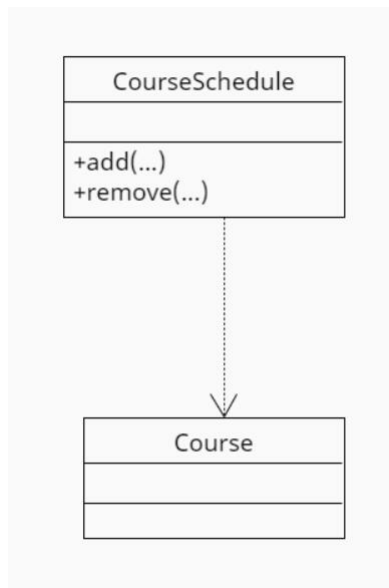
```

1 //4.14
2 #include <iostream>
3 using namespace std;
4
5 class Tree
6 {
7 public:
8     Tree(int age) : ages(age) {}
9     int age() { return ages; }
10    void grow(int years) { ages += years; }
11
12 private:
13     int ages;
14 };
15
16 int main()
17 {
18     Tree tree(1);
19     tree.grow(2);
20     cout << tree.age() << endl;
21     return 0;
22 }

```

4.15-4.18





4.19

```
1  #include <iostream>
2  using namespace std;
3
4  class CPU
5  {
6  private:
7      int f : 12;
8      int bits : 7;
9      int heshu : 3;
10     int cxc : 1;
11 };
12
13 int main()
14 {
15     cout << sizeof(CPU);
```

```
16 |     return 0;  
17 | }
```

```
ingw64/bin/gdb.exe' '--int  
er=mi'  
4  
PS E:\programs\cpps> █
```