

#### 4-1

**public** 定义类的外部接口；**private** 保护数据的安全性，只能被本类的成员函数访问。区别在于能否被外部访问。

#### 4-2

**protected** 继承过程中对产生的新类影响不同，它能够给派生类提供一些特殊的访问属性。

#### 4-3

构造函数：在定义对象时进行数据成员设置，即初始化，将对象初始化为一个特定的状态

析构函数：对象被删除前的一些清理工作

#### 4-4

数据成员可以公有，成员函数不能私有

#### 4-5

可以，**int a** 没有初始化，或没有执行完全相同的函数

#### 4-6

复制构造函数是一种特殊的构造函数，形参是本类对象的引用，可以使用一个已经存在的对象去初始化同类的一个新对象。

调用的情况有三种：（1）用类的一个对象去初始化另一个对象 （2）函数的形参是类的对象，值传递

（3）函数返回值是类的对象，在返回时会调用

#### 4-9

```
#include <iostream>

using namespace std;

class rectangle{

public:

    rectangle(int a,int b,int c,int d);

    void area();
```

```

private:

    int x1,y1,x2,y2;

};

rectangle::rectangle(int a,int b,int c,int d){

    x1 = a, y1 = b, x2 = c, y2 = d;

}

void rectangle::area(){

    int ans = (x2-x1)*(y2-y1);

    cout<<ans;

}

int main(){

    int a,b,c,d;

    cout<<"enter four number as coord:"<<endl;

    cin>>a>>b>>c>>d;

    rectangle rec1(a,b,c,d);

    rec1.area();

}

```

#### 4-13

```

#include <iostream>

using namespace std;

```

```
const float pi = 3.14159;

class circle{

public:

    circle(int x);

    void getArea();

private:

    int radius;

};

circle::circle(int x){

    radius = x;

}

void circle::getArea(){

    float ans = pi * radius * radius;

    cout<<"area is:"<<ans;

}

int main(){

    int x;

    cout<<"enter a integer as radius:";

    cin>>x;

    circle cir1(x);

    cir1.getArea();

}
```

测试结果:

```
oft-MIEngine-Error-ax5tbt2c.srt' --pid=Microsoft-MIEngine-Pid-k  
afr5kbo.dz1' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interpreter=mi  
,  
enter a integer as radius:4  
area is:50.2654  
D:\code_repository\codes\
```

#### 4-14

```
#include <iostream>  
  
using namespace std;  
  
const float pi = 3.14159;  
  
class tree{  
  
public:  
  
    tree(int x);  
  
    void grow(int years);  
  
    void age();  
  
private:  
  
    int ages;  
  
};  
  
tree::tree(int x){  
  
    ages=x;  
  
}  
  
void tree::grow(int years){  
  
    ages+=years;  
  
}
```

```

void tree::age(){

    cout<<"now the age is: "<<ages;

}

int main(){

    int x,y;

    cout<<"enter the origin age and year to add: ";

    cin>>x>>y;

    tree tree1(x);

    tree1.grow(y);

    tree1.age();

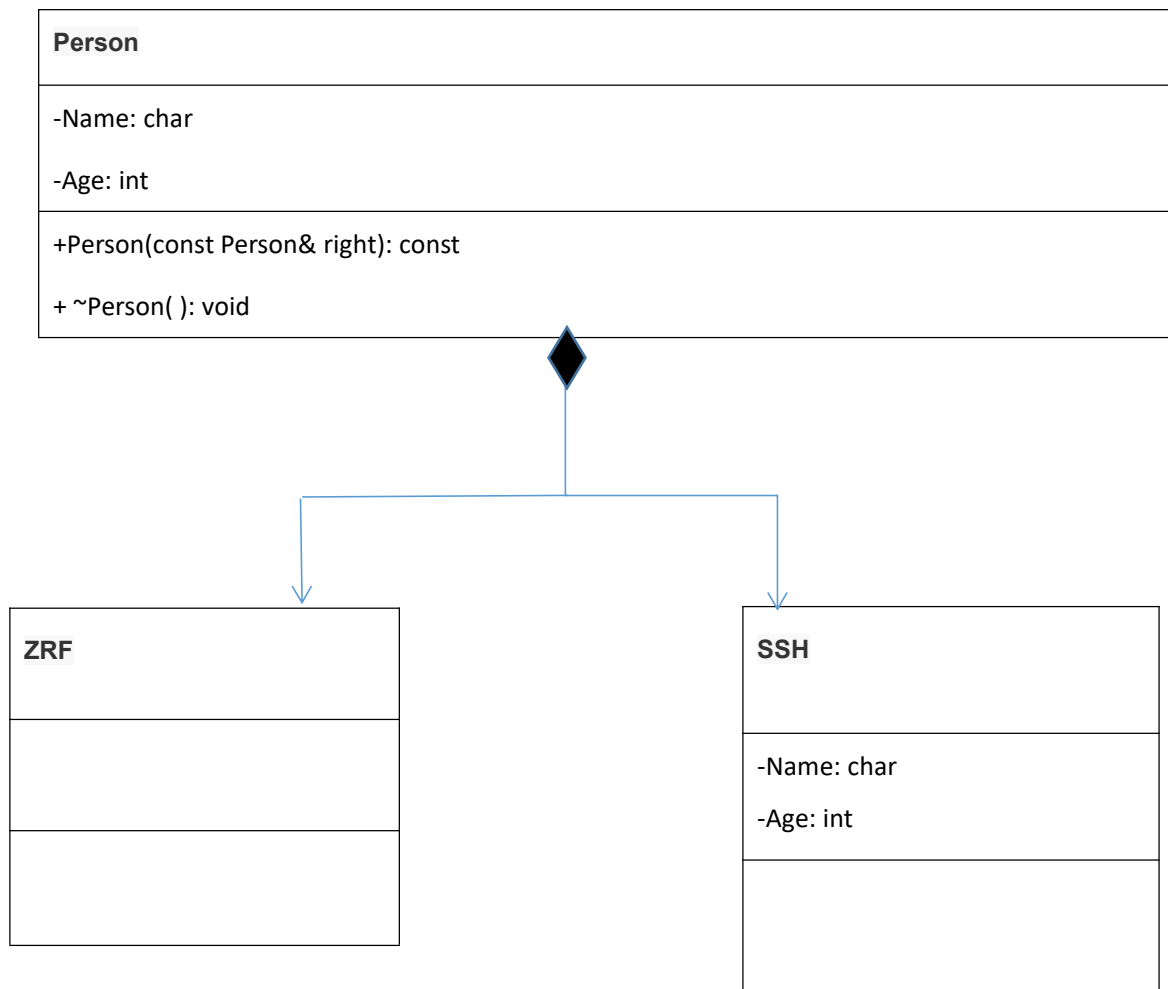
}

```

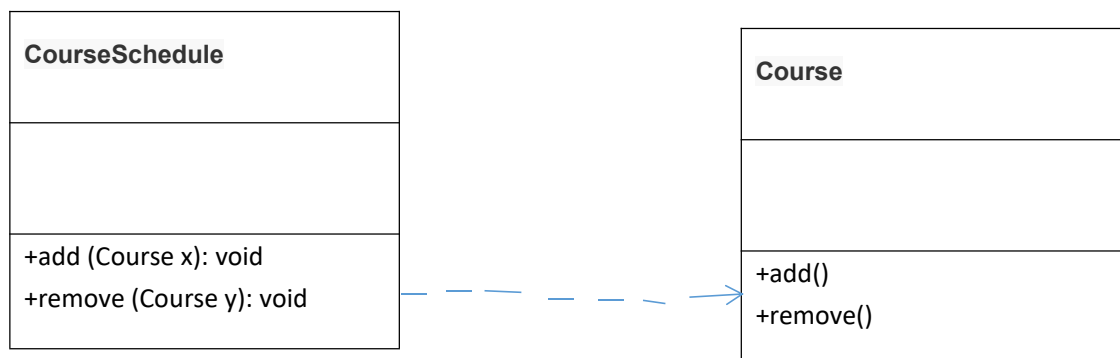
#### 4-15

<b>circle</b>
- radius: int
+circle(int x): void +getArea( ): void

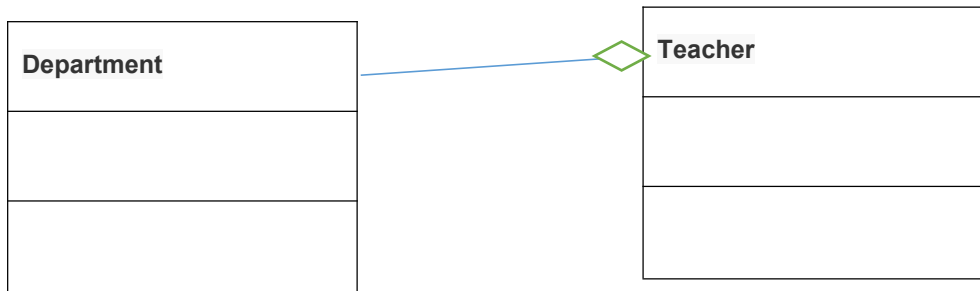
4-16



4-17



#### 4-18



#### 4-19

```
#include <iostream>

using namespace std;

class CPU{

private:

    unsigned int f:12;

    unsigned int len:1;

    unsigned int num:2;

    unsigned int check:1;

};

int main(){

    int l = sizeof(CPU);

    cout<<"size: "<<l;

}
```

```
sDebugLauncher.exe --stdin=Microsoft-MIEngine-In-fpje100b.yzt
'--stdout=Microsoft-MIEngine-Out-0y1anr3h.cxr' '--stderr=Micros
oft-MIEngine-Error-f031em01.sdm' '--pid=Microsoft-MIEngine-Pid-g
fzs40aq.n3s' '--dbgExe=D:\mingw64\bin\gdb.exe' '--interpreter=mi
'

size: 4
PS D:\code_repository\codes>
```