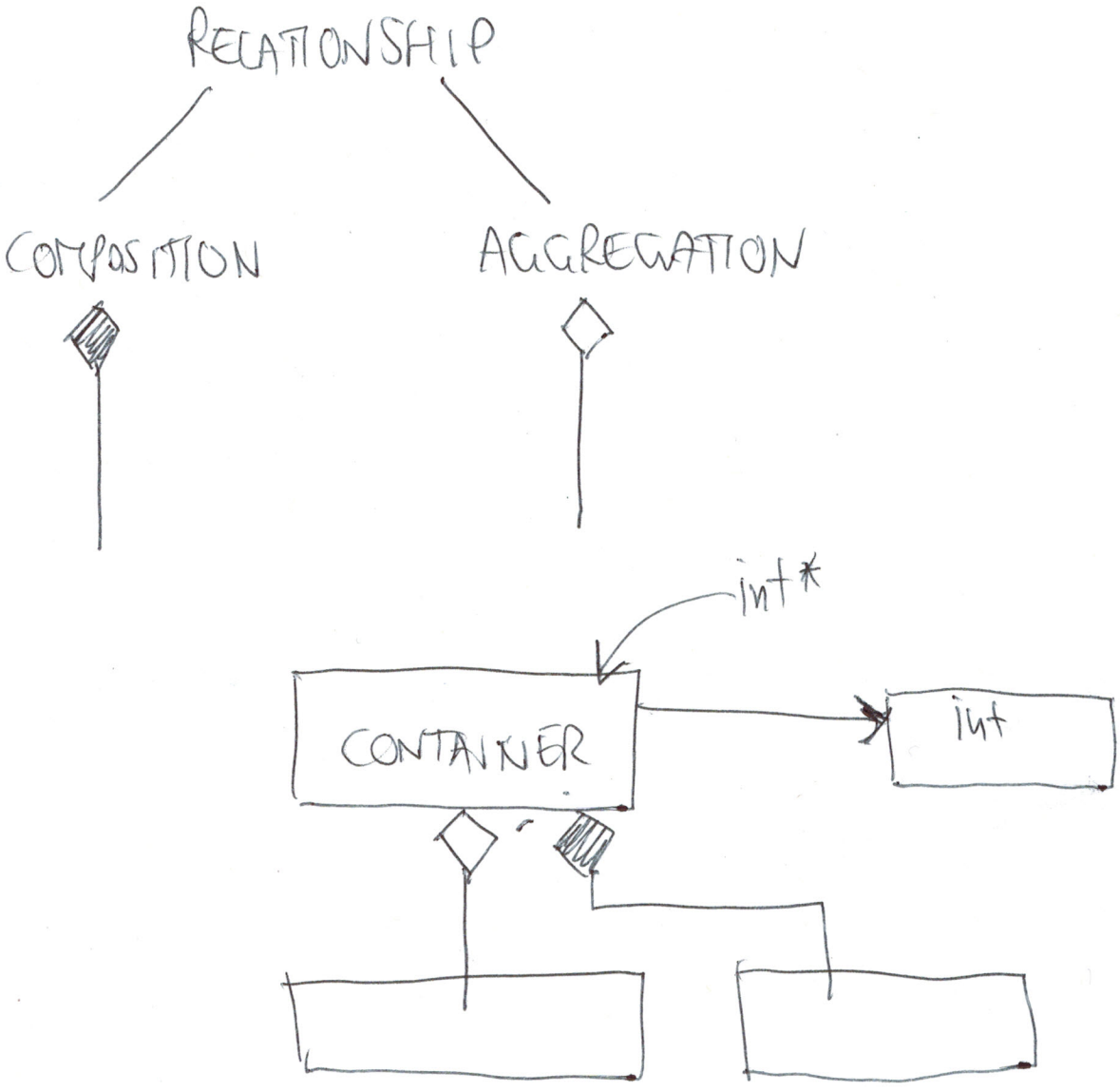
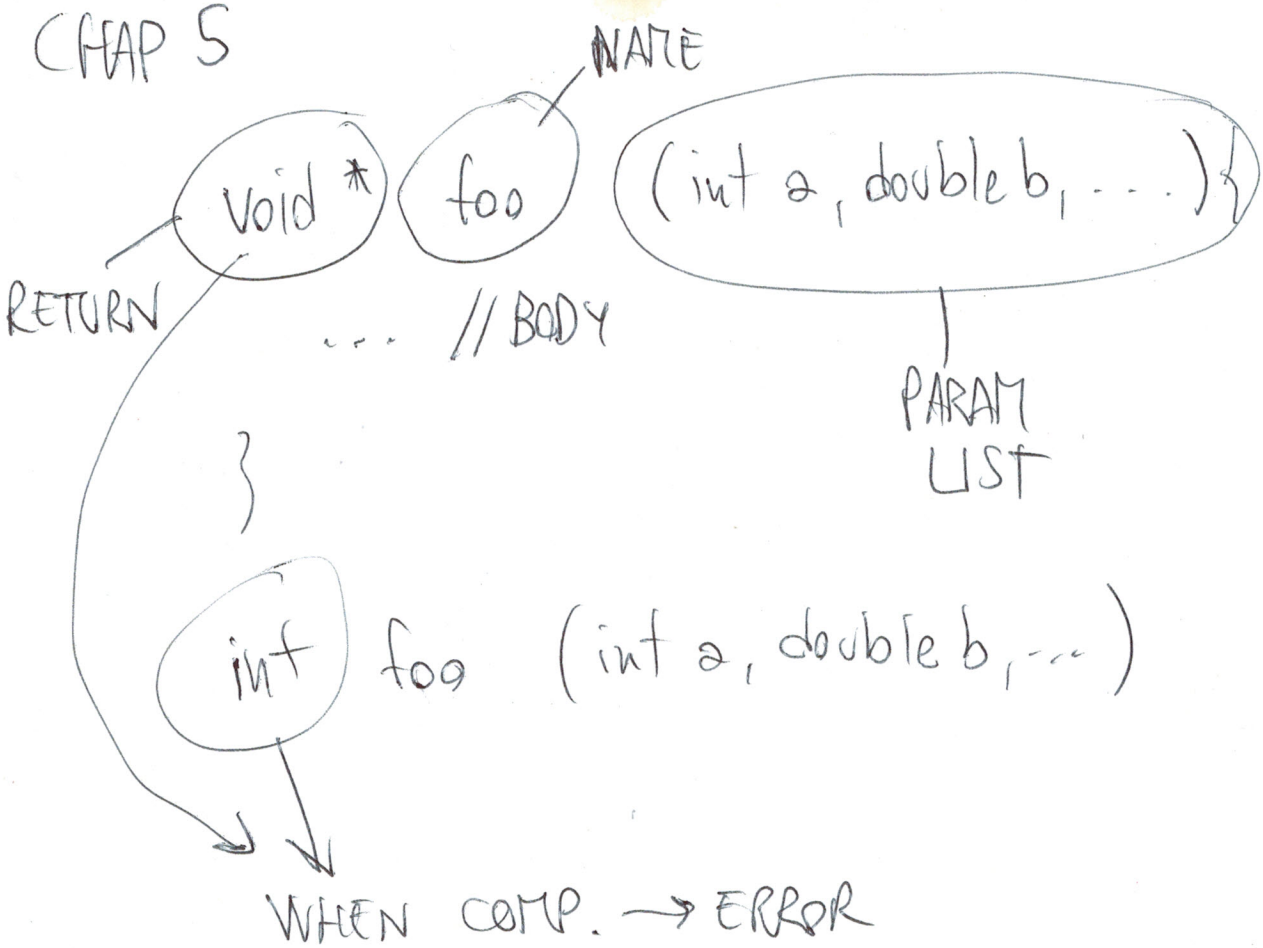
SEC 4.3



CHAP 5



```
void * foo (int a){
  ...
}
void * foo (float a){
  ...
}
```

SEC 5.2 OPTIONAL ARGS

ex 5.3

```
class Date {
public:
    Date(int d=0, int m=0, int y=0);
    ...
```

ex 5.5

```
Date d1;          // d=0, m=0, y=0
Date d2(15);      // d=15, m=0, y=0
Date d3(23,8);    // d=23, m=8, y=0
Date d4(19,11,2003); // d=19, m=11, y=2003
```

SEC 5.3 OPERATOR OVERLOADING

<< >> += -=

& *

YOU CANNOT OVERLOAD

- TERNARY OP
- SCOPE RESOLUTION ::
- MEMBER SELECT . AND .*

```
if (a>0)
   b=1;
else
   b=2;
```

→ TERNARY OP

```
b=(a>0)?1:2;
```

---

std::cout

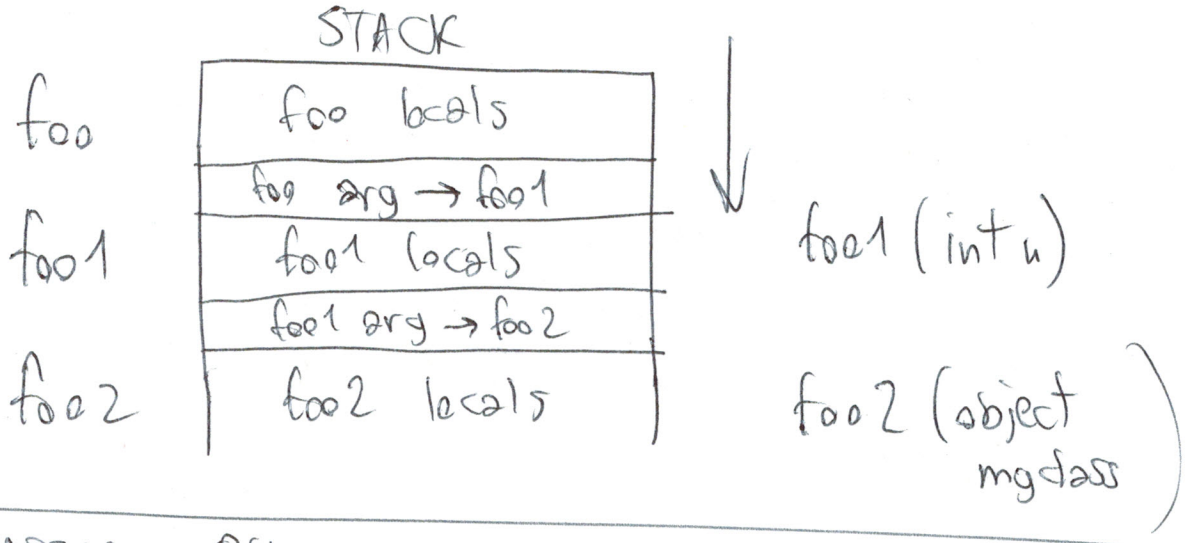
SCOPE RESOLUTION

```
class complex{
public:
    int Re;
    int Im;
};
```

```
complex comp;
comp.Re
```

Member Select

SEC 5.4 PASSING BY VALUE



SEC 5.5 PASSING BY REFERENCES

SKIP 5.6

5.7
5.8

```
int& maxi (int& x, int& y){
        return (x>y)? x: y;
}
```

SEC 5.11

```
printf("format string", . . . . )
```

VARIABLE ARGS

```
int printf (char * fmt, ...)
```

va_start (ap, p)

- ap: va_list type
- p: addres of last know argument (the argument before the variable list)

va_end (ap)

---

va_arg (ap, typename)

5.20 example show how to use it!