



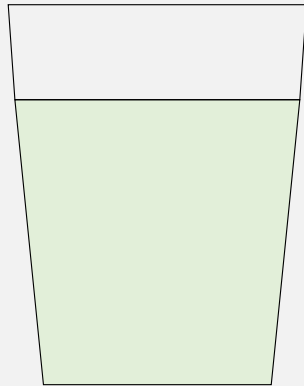
THE SWAPPING PROBLEM

Version 3

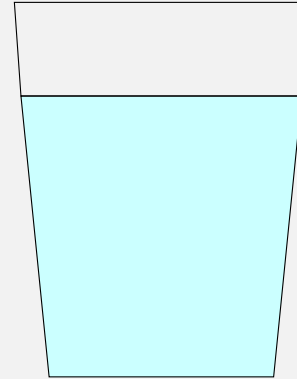


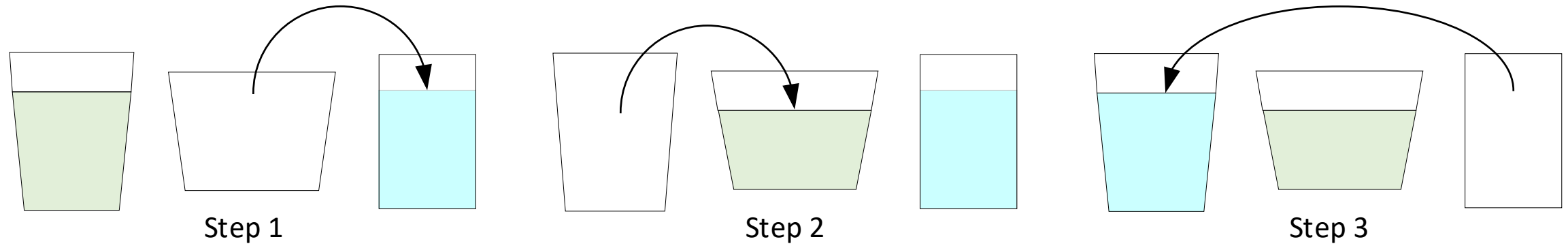
THE PROBLEM

INITIAL STATE



GOAL (SOLUTION) STATE





THE SOLUTION



PROGRAMMING THE SOLUTION

GENERAL SOLUTION

- `T temp = g2;` `// step 1`
- `g2 = g1;` `// step 2`
- `g1 = temp;` `// step 3`

EXAMPLE

```
struct student
{
    int    id;
    string name;
    double gpa;
};

student temp = s2;
s2 = s1;
s1 = temp;
```



PASSING FUNCTION ARGUMENTS

CHOICES

- `void swap(T p1, T p2);`
- Pass-by-value
- Pass-by-reference
- Pass-by-pointer



PASSING FUNCTION ARGUMENTS

CHOICES

- `void swap(T p1, T p2);`
- Pass-by-value
- Pass-by-reference
- Pass-by-pointer

REASONING

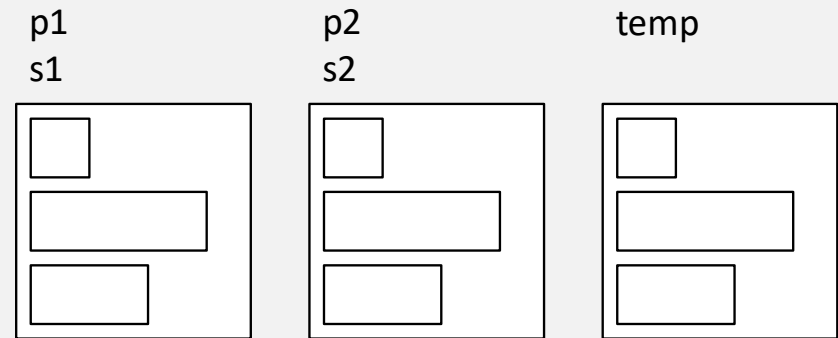
- Function must change two arguments
 - Argument passing must be INOUT
- Pass-by-value: IN only
- Pass-by-reference: INOUT
- Pass-by-pointer: INOUT



PASS-BY-REFERENCE

```
void swap(student& p1, student& p2)
{
    student temp = p2;
    p2 = p1;
    p1 = temp;
}
```

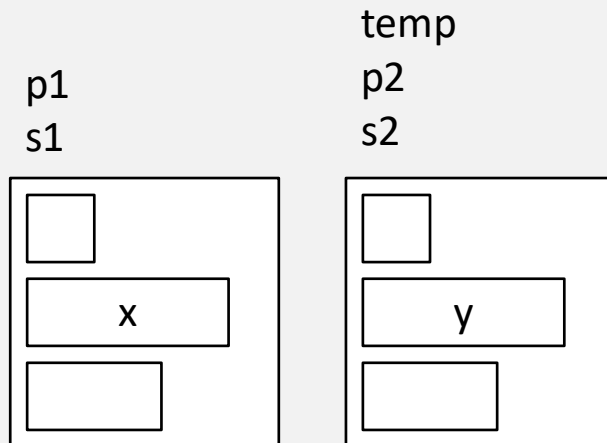
```
student s1 = { 123, "dilbert", 3.0 };
student s2 = { 987, "alice", 4.0 };
swap(s1, s2);
```





PASS-BY-REFERENCE: FAIL

student& temp = p2;

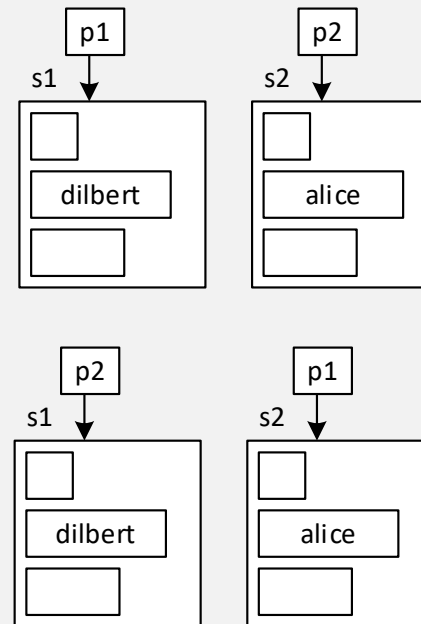


	p1	p2	temp
	x	y	
temp& = p2	x	y	y
p2 = p1	x	x	x
p1 = temp	x	x	x

PASS-BY-POINTER (I)

```
void swap(student* p1, student* p2)
{
    student temp = *p2;
    *p2 = *p1;
    *p1 = temp;
}
```

```
student s1 = { 123, "dilbert", 3.0 };
student s2 = { 987, "alice", 4.0 };
swap(&s1, &s2);
```





PASS-BY-POINTER (2)

```
void swap(student* p1, student* p2)
{
    student temp = *p2;
    *p2 = *p1;
    *p1 = temp;
}
```

```
student* s1 = new student { 123, "dilbert", 3.0 };
student* s2 = new student { 987, "alice", 4.0 };
swap(s1, s2);
```