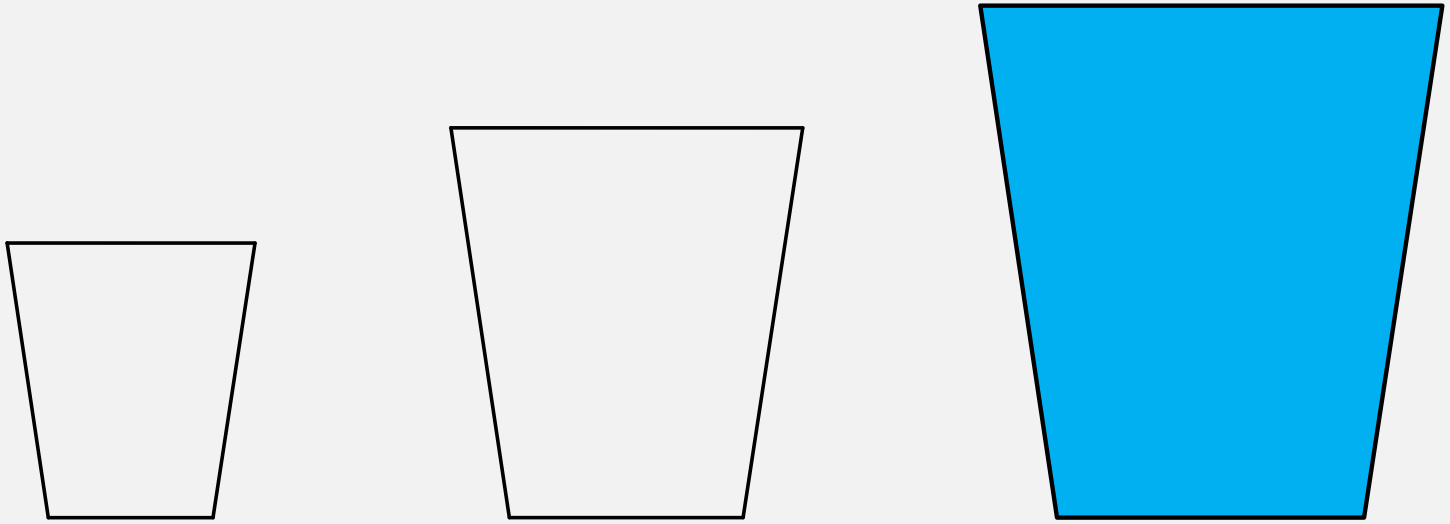
POURING PUZZLE

Objects and Member Functions

Delroy A. Brinkerhoff

THE PROBLEM

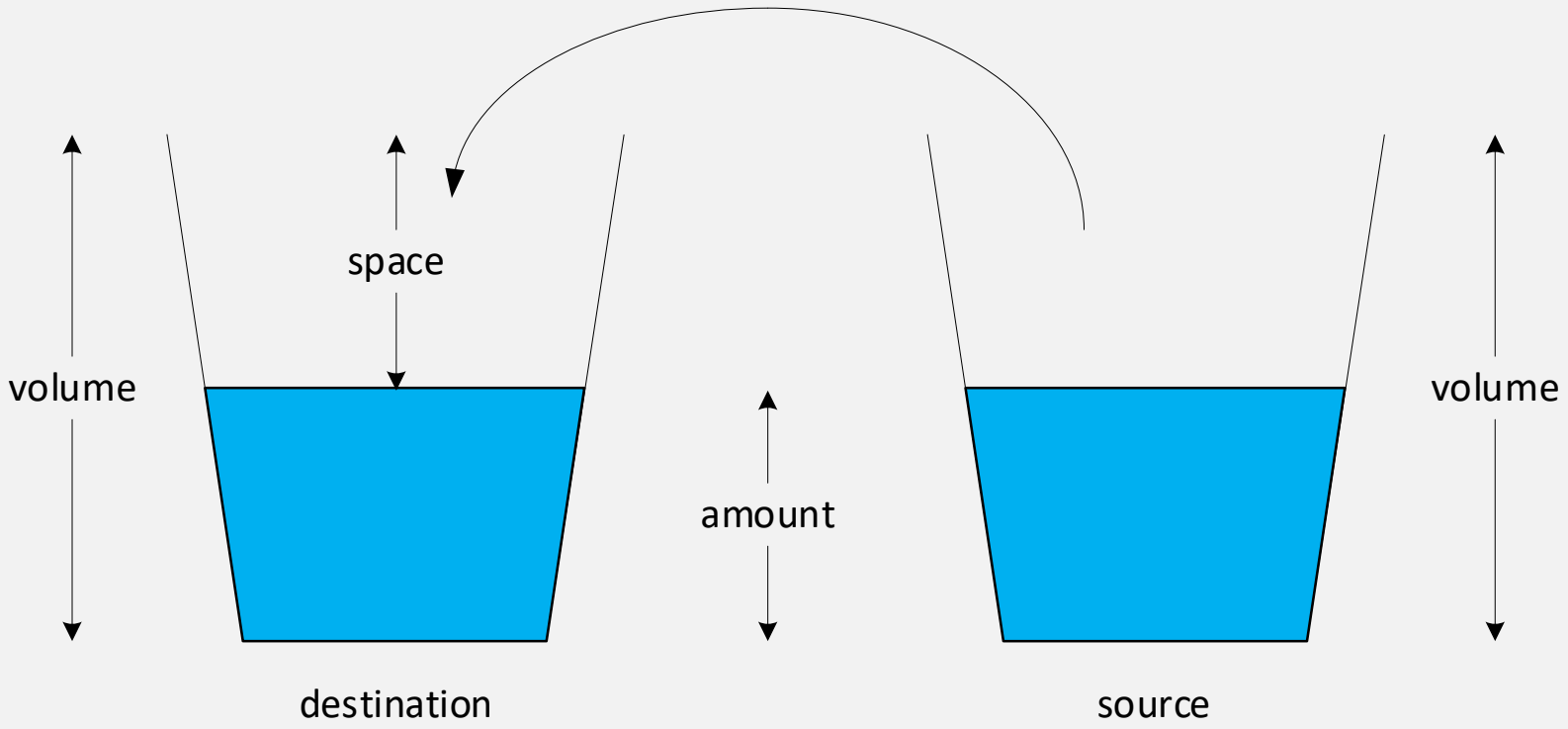


3 oz.

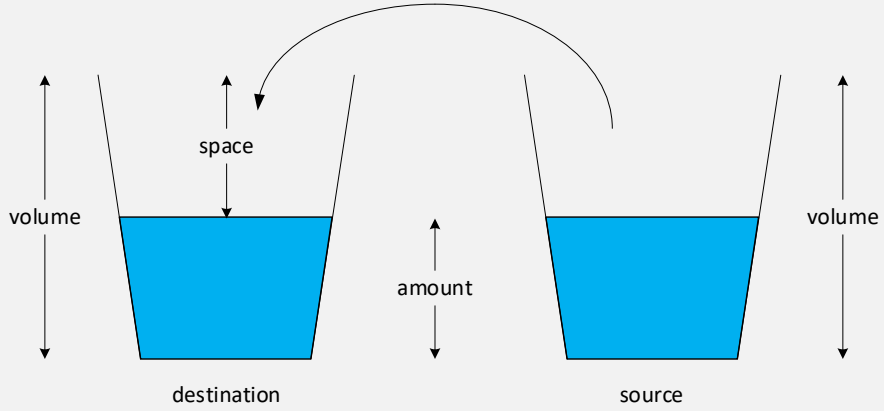
5 oz.

8 oz.

THE POUR OPERATION:
`destination.pour(source)`

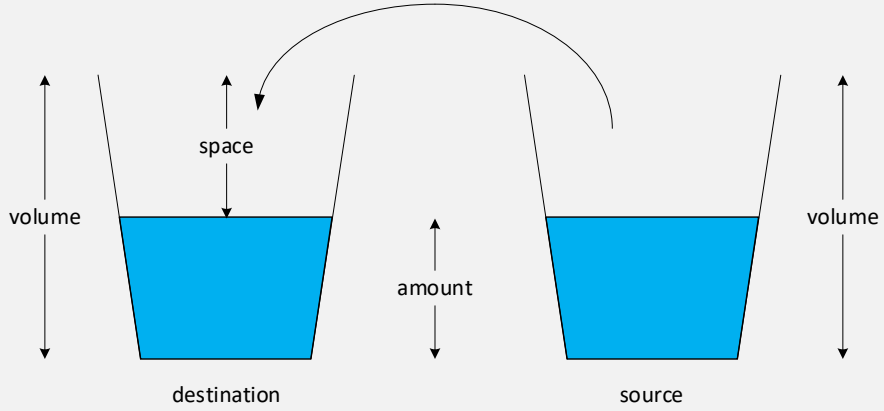


POURING ALGORITHM (VERSION 1):
destination.pour(source)




```
int    space = volume - amount;

if (space < source.amount)
{
        source.amount -= space
        amount = volume
}
else
{
        amount += source.amount
        source.amount = 0
}
```

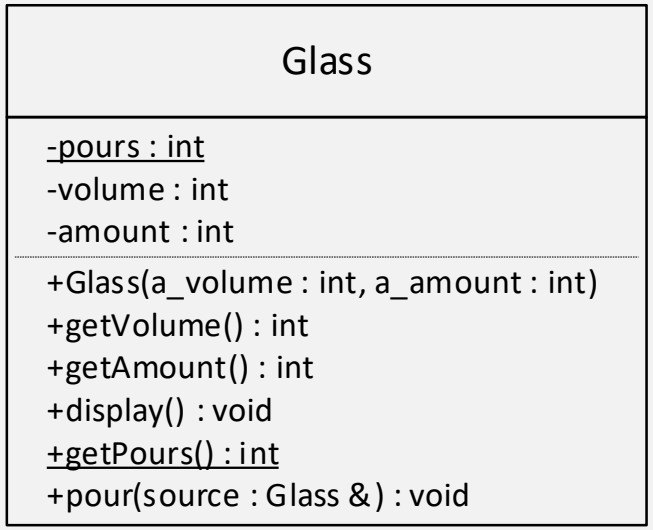
POURING ALGORITHM (VERSION 2):
destination.pour(source)



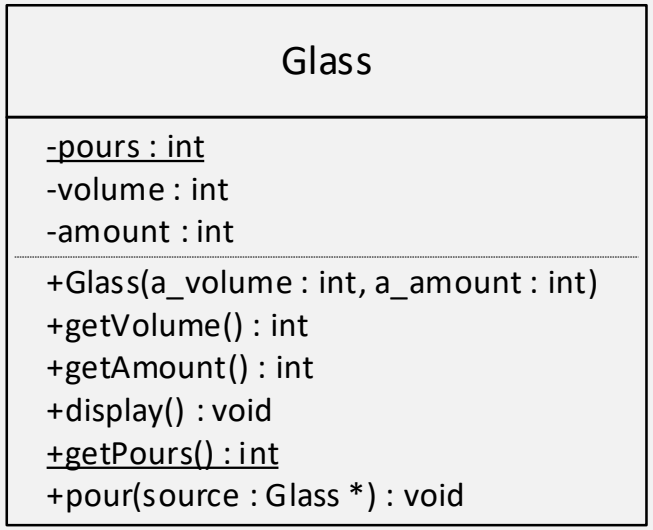

```
int    space = volume - amount;

transfer = min(space, source.amount)
amount += transfer
source.amount -= transfer
```

PASSING THE "SOURCE" GLASS

PASS BY REFERENCE



PASS BY POINTER



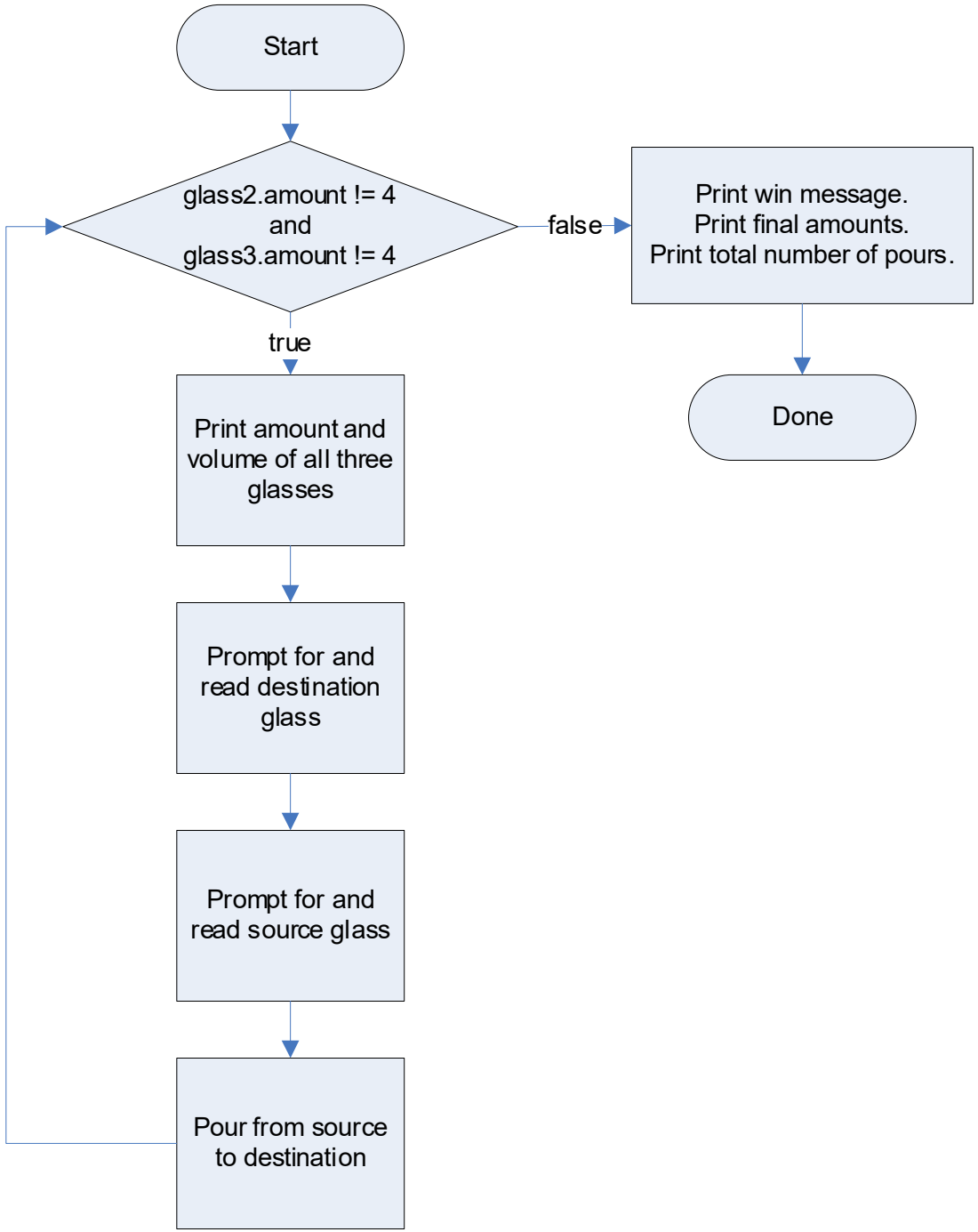
MANAGING THE GLASSES: SEPARATE OBJECTS

```
Glass g1(3, 0);
Glass g2(5, 0);
Glass g3(8, 8);
    . . . .
g1.pour(g2);
g2.pour(g1);
g1.pour(g3);
g3.pour(g1);
g2.pour(g3);
g3.pour(g2);
```

```
while (the puzzle is not solved)
{
    int  source = user input;
    int  destination = user input;
              . . . .
    if (destination == 1 && source == 2)
        g1.pour(g2);
    else if (destination == 2 && source == 1)
        g2.pour(g1);
              . . .
}
```

MANAGING THE GLASSES: ARRAY OF OBJECTS

```
Glass  glasses[3];

while (the puzzle is not solved)
{
        int    source = user input;
        int    destination = user input;
               . . . .
        glasses[destination].pour(glasses[source]);
}
```



THE PUZZLE LOGIC