

# Homework 6

## Problem 1

a)

```
baz  
bar  
1
```

b)

```
baz  
bar  
4
```

c)

```
baz  
bar  
1 - object is immutable so doesn't change
```

d)

```
baz  
bar  
1
```

## Problem 2

Using the Optional struct:

- consumer of API has to explicitly check whether a result is valid each time
- dynamically allocating new int(i) requires a destructor to deallocate it once it's no longer needed

Using the native exception handling:

- performance overhead from exception call, especially if this function is used often
- cleaner code - easier to read
- consumer of API needs to unify it with try and except block
- risk of runtime errors if forget to check exception at any layer

If a nullptr isn't desired in the code at any point, then it would make more sense to use an exception than have the user manually check if the result is valid and then deallocate the integer than just returning the integer in the function

## Problem 3

a)

```
catch 2
i'm done
that's what I say
really done
```

b)

```
catch 1
hurray
i'm done
```

```
that's what I say  
really done
```

**d)**

```
catch 3
```