J Bolton Chapter 7

Programming Language Design

7.4, 7.8, 7.16 (design only)

7.4

Arithmetic on integers, floats, ascii values, pointers

7.8

C:

```
#include <stdio.h>
#include <math.h>

int main() {
    int i = pow(2, 30) + 65537;
    printf("Initial int value: %d\n", i);
    float f = i + 1.0;
    printf("Float value: %f\n", f);
    int i2 = f - 1;
    printf("Back to int value: %d\n", i2);
    return 0;
}
```

Output:

```
Initial int value: 1073807361
Float value: 1073807360.000000
Back to int value: 1073807359
```

Java:

```
public class ifconv {
    public static void main(String[] args) {
        int i = (int) java.lang.Math.pow(2, 30) + 65534;
        System.out.println("Initial int: " + i);
        float f = i + (float) 1.0;
        System.out.println("Float value: " + f);
        int i2 = (int) f - 1;
        System.out.println("New int value: " + i2);
    }
}
```

Output:

```
Initial int: 1073807358
Float value: 1.07380736E9
New int value: 1073807359
```

7.16

```
// (a)
public Stack(int size) { stack = new int[size]; }

public static void main(String[] args) {
     Stack test = new Stack(10);
     test.push(9);
     test.push(1);
     test.pop();
     test.pop();
     test.pop();  // should throw exception
}

// (b)
public int push(int n) throws StackOverflowException {
     if (top >= stack.size()) {
          throw new StackOverflowException(“stack is full”);
     }
     return stack[++top] = n;
}

// (c)
// I'm not seeing how assert would be useful in the constructor..
public Stack(int size) { stack = new int[size]; }

public static void main(String[] args) {
     Stack test = new Stack(10);
     assert top < stack.size() : top;
     test.push(9);
     test.push(1);
     test.pop();
     test.pop();
     assert top > 0 : top;
     test.pop();  // should not reach here
}
```