JAVA CODE:

```
//Copyright 2025 Abby Holdcraft
public class FizzBuzz {
    public static int SIZE = 100;
    public static void main(String[] args) {
        System.out.println("Playing FizzBuzz...");

        for(int i=1;i<=SIZE;i++) {
            boolean has_printed = false;
            if(i%3==0) {
                System.out.print("FIZZ");
                has_printed = true;
            }
            if(i%5==0){
                System.out.print("BUZZ");
                has_printed = true;
            }
            if(!has_printed)
                System.out.print(i);
            System.out.println();
        }
    }
}
```

JAVA OUTPUT:

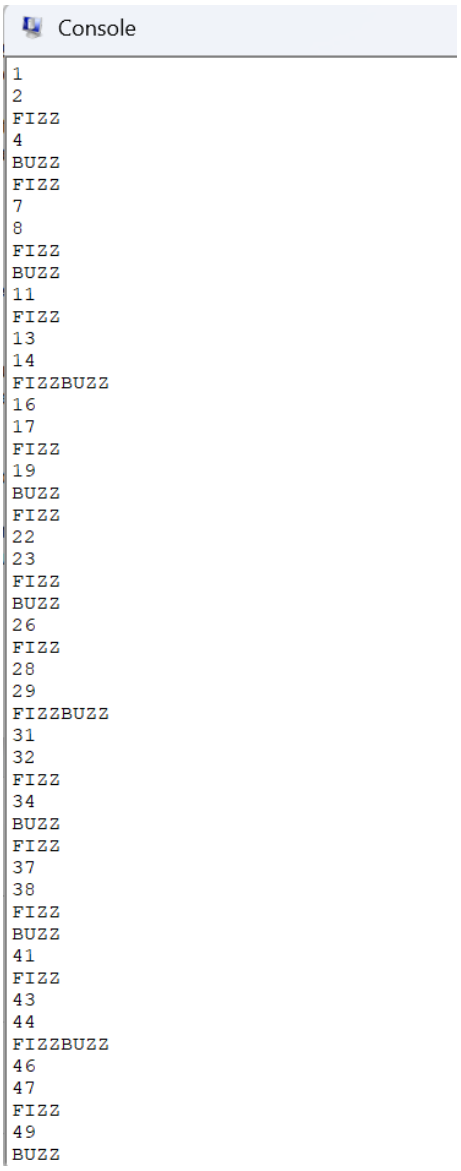
```
Playing FizzBuzz...
1
2
FIZZ
4
BUZZ
FIZZ
7
8
FIZZ
BUZZ
11
FIZZ
13
14
FIZZBUZZ
16
17
FIZZ
19
BUZZ
FIZZ
22
23
FIZZ
BUZZ
26
FIZZ
28
29
FIZZBUZZ
31
32
FIZZ
34
BUZZ
FIZZ
37
38
FIZZ
BUZZ
41
FIZZ
43
43
44
FIZZBUZZ
46
47
FIZZ
49
BUZZ
FIZZ
52
53
FIZZ
BUZZ
56
FIZZ
58
59
FIZZBUZZ
61
62
FIZZ
64
BUZZ
FIZZ
67
68
FIZZ
BUZZ
71
FIZZ
73
74
FIZZBUZZ
76
77
FIZZ
79
BUZZ
FIZZ
82
83
FIZZ
BUZZ
86
67
68
FIZZ
BUZZ
71
FIZZ
73
74
FIZZBUZZ
76
77
FIZZ
79
BUZZ
FIZZ
82
83
FIZZ
BUZZ
86
FIZZ
88
89
FIZZBUZZ
91
92
FIZZ
94
BUZZ
FIZZ
97
98
FIZZ
BUZZ
```

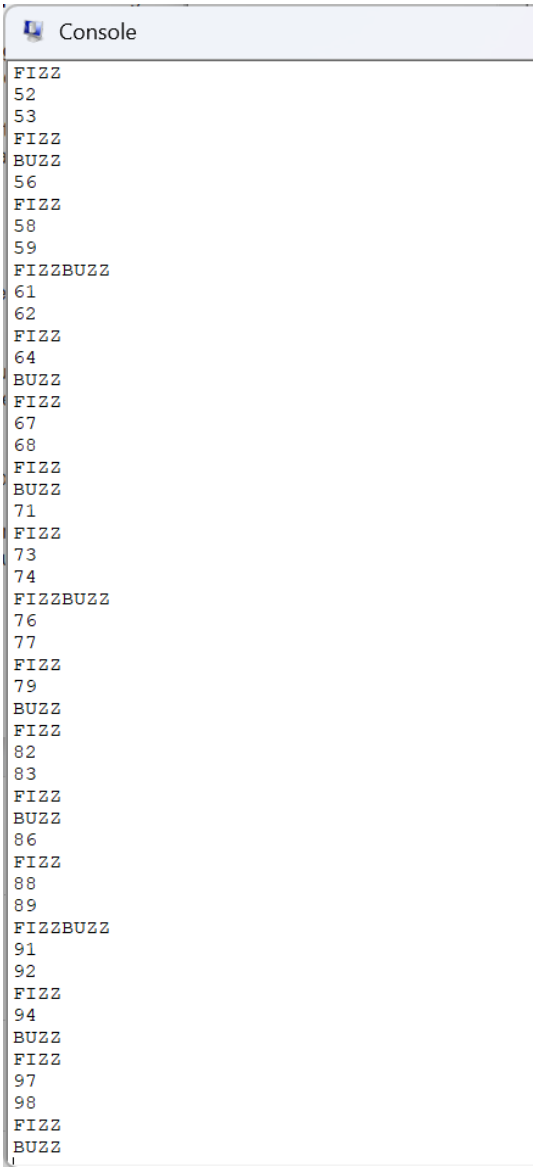
MIPS CODE:

```
# Copyright 2025 Abby Holdcraft
    .data
size: .word 100
new_line: .asciiz "\n"
fizz: .asciiz "FIZZ"
buzz: .asciiz "BUZZ"

    .text
main:
    li $t0, 1                   # i = 1
    lw $t1, size    # end when i > size

    li $t2, 3                   # divide by 3
    li $t3, 5                   # divide by 5
    li $t5, 0                   # has_printed = false

    j check_fizz   # begin printing FizzBuzz

loop_continue:
    la $a0, new_line        # load new_line into a0
    li $v0, 4                   # print string
    syscall

    addi $t0, $t0, 1          # i++

    li $t5, 0                   # has_printed = false

    bgt $t0, $t1, exit        # if i>size, exit
    j check_fizz   # else, restart loop

check_fizz:
    # check if divisible by 3
    div $t0, $t2        # divide i by 3
    mfhi $t4           # store remainder in t4
    beq $t4, 0, print_fizz      # if remainder == 0, print fizz
    j check_buzz              # else, check buzz

check_buzz:
    # check if divisible by 5
    div $t0, $t3        # divide i by 5
    mfhi $t4           # store remainder in t4
    beq $t4, 0, print_buzz      # if remainder == 0, print buzz
    beq $t5, 0, print_number        # if has_printed == false, print i
```

```
    j loop_continue                 # else, continue loop

print_fizz:
    la $a0, fizz        # load buzz into a0
    li $v0, 4           # print string
    syscall

    li $t5, 1                   # has_printed = true

    j check_buzz # check buzz

print_buzz:
    la $a0, buzz   # load buzz into a0
    li $v0, 4                   # print string
    syscall

    li $t5, 1                   # has_printed = true

    j loop_continue         # return to loop

print_number:
    move $a0, $t0           # copy value of t0 into a0
    li $v0, 1                   # print int
    syscall

    j loop_continue         # return to loop

exit:
    li $v0, 10
    syscall
```

MIPS OUTPUT:


Console
1
2
FIZZ
4
BUZZ
FIZZ
7
8
FIZZ
BUZZ
11
FIZZ
13
14
FIZZBUZZ
16
17
FIZZ
19
BUZZ
FIZZ
22
23
FIZZ
BUZZ
26
FIZZ
28
29
FIZZBUZZ
31
32
FIZZ
34
BUZZ
FIZZ
37
38
FIZZ
BUZZ
41
FIZZ
43
44
FIZZBUZZ
46
47
FIZZ
49
BUZZ


Console
FIZZ
52
53
FIZZ
BUZZ
56
FIZZ
58
59
FIZZBUZZ
61
62
FIZZ
64
BUZZ
FIZZ
67
68
FIZZ
BUZZ
71
FIZZ
73
74
FIZZBUZZ
76
77
FIZZ
79
BUZZ
FIZZ
82
83
FIZZ
BUZZ
86
FIZZ
88
89
FIZZBUZZ
91
92
FIZZ
94
BUZZ
FIZZ
97
98
FIZZ
BUZZ