# **CS 410 Binary to C++ With Security Vulnerabilities Activity Template**

**Step 1: Convert the binary file to assembly code.**

The binary file has been disassembled using objdump, resulting in the assembly code provided in assignment6\_1.asm.

**Step 2: Explain the functionality of the blocks of assembly code.**

|  |  |
| --- | --- |
| **Blocks of Assembly Code** | **Explanation of Functionality** |
| Program Entry (0xf80-0xfaa) | Sets up program environment, configures stack, and initializes runtime |
| Input Validation (0x108a-0x10d5) | Checks std::cin for failures and clears error flags if needed |
| Bounds Checking (0x10d6-0x10fc) | Verifies if numbers are within valid integer range |
| Menu Display (0x10fd-0x1175) | Shows calculator menu options and prompts for user input |
| Calculator Logic (0x1176-0x171c) | Handles arithmetic operations and checks for overflows |

**Step 3: Convert the assembly code to binary**Here are the key assembly instructions converted to their binary representation:

Program Entry:

f80: 31 ed ; xor %ebp,%ebp

f82: 49 89 d1 ; mov %rdx,%r9

f85: 5e ; pop %rsi

f86: 48 89 e2 ; mov %rsp,%rdx

Input Validation:

108a: 55 ; push %rbp

108b: 48 89 e5 ; mov %rsp,%rbp

108e: 48 8d 3d bb 10 20 00 ; lea 0x2010bb(%rip),%rdi

1095: e8 a6 fd ff ff ; callq e40

Bounds Checking:

10d6: 55 ; push %rbp

10d7: 48 89 e5 ; mov %rsp,%rbp

10da: 89 7d fc ; mov %edi,-0x4(%rbp)

10dd: 81 7d fc 00 00 00 80 ; cmpl $0x80000000,-0x4(%rbp)

Menu Display:

10fd: 55 ; push %rbp

10fe: 48 89 e5 ; mov %rsp,%rbp

1101: 48 8d 35 11 07 00 00 ; lea 0x711(%rip),%rsi

1108: 48 8d 3d 11 0f 20 00 ; lea 0x200f11(%rip),%rdi

Main Function:

1176: 55 ; push %rbp

1177: 48 89 e5 ; mov %rsp,%rbp

117a: 41 55 ; push %r13

117c: 41 54 ; push %r12

117e: 53 ; push %rbx

Each line shows:

* Memory address (in hex)
* Machine code bytes
* Corresponding assembly instruction

The machine code bytes represent the actual binary that the CPU executes. For example:

* 55 is the binary for push %rbp
* 48 89 e5 is the binary for mov %rsp,%rbp
* 31 ed is the binary for xor %ebp,%ebp

**Step 4: Convert the assembly code to C++ code.**

|  |  |
| --- | --- |
| **Blocks of Assembly Code** | **C++ Code** |
| Input Validation Block | bool validateInput() { if (std::cin.fail()) { std::cin.clear(); std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n'); return false; } return true; } |
| Bounds Check Block | bool isWithinBounds(int value) { return (value > INT\_MIN && value < INT\_MAX); } |
| Menu Display Block | void DisplayMenu() { std::cout << "Calculator Menu\n1. Subtraction\n2. Addition\n3. Division\n4. Exit\nEnter your choice: "; } |
| Main Function Block | int main() { int choice = 0; int num1 = 0; int num2 = 0; while (choice != 4) { DisplayMenu(); /\* Menu logic \*/ } return 0; } |
| Exception Handler Block | try { /\* Operation code \*/ } catch (const std::exception& e) { std::cout << "Error: " << e. |