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

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

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

| **Blocks of Assembly Code** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp  sub $0x20,%rsp  mov %fs:0x28,%rax  mov %rax,-0x8(%rbp)  xor %eax,%eax  movl $0x0,-0x14(%rbp)  mov -0x14(%rbp),%eax  cmp $0x5,%eax  je d02 <main+0x28f> | Start prologue  End prologue  Loops begins until %eax and 5 is true  Jumps to d02 <main+0x28f> |
| lea 0x3a5(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  lea 0x201575(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x3a4(%rip),%rsi # e5b <\_ZStL19piecewise\_construct+0x57>  lea 0x201562(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x39c(%rip),%rsi # e66 <\_ZStL19piecewise\_construct+0x62>  lea 0x20154f(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x399(%rip),%rsi # e76 <\_ZStL19piecewise\_construct+0x72>  lea 0x20153c(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x396(%rip),%rsi # e86 <\_ZStL19piecewise\_construct+0x82>  lea 0x201529(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x346(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  lea 0x201516(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | The DisplayMenu function prints out  ----------------  - 1) Add -  - 2) Subtract -  - 3) Multiply -  - 4) Exit -  ---------------- |
| lea 0x396(%rip),%rsi # e86 <\_ZStL19piecewise\_construct+0x82>  lea 0x201529(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Loads and calls the menu selection |
| lea 0x346(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  lea 0x201516(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Loads and calls to retrieve first input |
| lea -0x14(%rbp),%rax  mov %rax,%rsi  lea 0x201623(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt> | Loads and calls to retrieve second input |
| mov -0x14(%rbp),%eax  cmp $0x1,%eax  jne bc3 <main+0x150>  lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201604(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt>  mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 870 <\_ZNSirsERi@plt>  mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x2014c1(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 8d0 <\_ZNSolsEi@plt>  lea 0x327(%rip),%rsi # e92 <\_ZStL19piecewise\_construct+0x8e>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  lea 0x30c(%rip),%rsi # e96 <\_ZStL19piecewise\_construct+0x92>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  sub %eax,%edx  mov %edx,%eax  mov %eax,%esi  mov %rcx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x20141d(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq 8a0 <\_ZNSolsEPFRSoS\_E@plt>  jmpq a91 <main+0x1e> | Subtracts the first and second input  Calculates total |
| mov -0x14(%rbp),%eax  cmp $0x2,%eax  jne c62 <main+0x1ef>  lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201563(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt>  mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 870 <\_ZNSirsERi@plt>  mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201420(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 8d0 <\_ZNSolsEi@plt>  lea 0x286(%rip),%rsi # e92 <\_ZStL19piecewise\_construct+0x8e>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  lea 0x26b(%rip),%rsi # e96 <\_ZStL19piecewise\_construct+0x92>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  add %edx,%eax  mov %eax,%esi  mov %rcx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x20137e(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq 8a0 <\_ZNSolsEPFRSoS\_E@plt>  jmpq a91 <main+0x1e> | Adds the first and second input  Calculates total |
| mov -0x14(%rbp),%eax  cmp $0x3,%eax  jne a91 <main+0x1e>  lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x2014c4(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt>  mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 870 <\_ZNSirsERi@plt>  mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201381(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 8d0 <\_ZNSolsEi@plt>  lea 0x1e7(%rip),%rsi # e92 <\_ZStL19piecewise\_construct+0x8e>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  lea 0x1cc(%rip),%rsi # e96 <\_ZStL19piecewise\_construct+0x92>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rcx  mov -0x10(%rbp),%eax  mov -0xc(%rbp),%esi  cltd  idiv %esi  mov %eax,%esi  mov %rcx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x2012de(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq 8a0 <\_ZNSolsEPFRSoS\_E@plt>  jmpq a91 <main+0x1e> | Divides the first and second input  Calculates total |
| mov $0x0,%eax | Resets the values |
| mov -0x8(%rbp),%rcx  xor %fs:0x28,%rcx  je d1b <main+0x2a8> | Loops back to the beginning of the code |
| callq 8b0 <\_\_stack\_chk\_fail@plt>  leaveq  retq | Calls and exits program |
|  |  |

**Step 3:** Convert the assembly code to binary.

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

| **Blocks of Assembly Code** | **C++ Code** |
| --- | --- |
| mov -0x14(%rbp),%eax  cmp $0x5,%eax  je d02 <main+0x28f> | cin >> choice  while (choice < 5)) {  int choice = 0; |
| lea 0x3a5(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  lea 0x201575(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x3a4(%rip),%rsi # e5b <\_ZStL19piecewise\_construct+0x57>  lea 0x201562(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x39c(%rip),%rsi # e66 <\_ZStL19piecewise\_construct+0x62>  lea 0x20154f(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x399(%rip),%rsi # e76 <\_ZStL19piecewise\_construct+0x72>  lea 0x20153c(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x396(%rip),%rsi # e86 <\_ZStL19piecewise\_construct+0x82>  lea 0x201529(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x346(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  lea 0x201516(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | void DisplayMenu() {  cout << "----------------" << endl;  cout << "- 1) Add -" << endl;  cout << "- 2) Subtract -" << endl;  cout << "- 3) Multiply -" << endl;  cout << "- 4) Exit -" << endl;  cout << "----------------" << endl; |
| lea 0x396(%rip),%rsi # e86 <\_ZStL19piecewise\_construct+0x82>  lea 0x201529(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | DisplayMenu() |
| lea 0x346(%rip),%rsi # e49 <\_ZStL19piecewise\_construct+0x45>  lea 0x201516(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | std::cin >> num1; |
| lea -0x14(%rbp),%rax  mov %rax,%rsi  lea 0x201623(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt> | std::cin >> num2; |
| mov -0x14(%rbp),%eax  cmp $0x1,%eax  jne bc3 <main+0x150>  lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201604(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt>  mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 870 <\_ZNSirsERi@plt>  mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x2014c1(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 8d0 <\_ZNSolsEi@plt>  lea 0x327(%rip),%rsi # e92 <\_ZStL19piecewise\_construct+0x8e>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  lea 0x30c(%rip),%rsi # e96 <\_ZStL19piecewise\_construct+0x92>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  sub %eax,%edx  mov %edx,%eax  mov %eax,%esi  mov %rcx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x20141d(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq 8a0 <\_ZNSolsEPFRSoS\_E@plt>  jmpq a91 <main+0x1e> | if (choice == 1) {  std::cout << num1 << “-“ << num2 << “=” num1 -num2 << std::endl; } |
| mov -0x14(%rbp),%eax  cmp $0x2,%eax  jne c62 <main+0x1ef>  lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x201563(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt>  mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 870 <\_ZNSirsERi@plt>  mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201420(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 8d0 <\_ZNSolsEi@plt>  lea 0x286(%rip),%rsi # e92 <\_ZStL19piecewise\_construct+0x8e>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  lea 0x26b(%rip),%rsi # e96 <\_ZStL19piecewise\_construct+0x92>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rcx  mov -0x10(%rbp),%edx  mov -0xc(%rbp),%eax  add %edx,%eax  mov %eax,%esi  mov %rcx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x20137e(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq 8a0 <\_ZNSolsEPFRSoS\_E@plt>  jmpq a91 <main+0x1e> | else if (choice == 2) {  std::cout << num1 << “+“ << num2 << “=” num1 +num2 << std::endl; } |
| mov -0x14(%rbp),%eax  cmp $0x3,%eax  jne a91 <main+0x1e>  lea -0x10(%rbp),%rax  mov %rax,%rsi  lea 0x2014c4(%rip),%rdi # 202140 <\_ZSt3cin@@GLIBCXX\_3.4>  callq 870 <\_ZNSirsERi@plt>  mov %rax,%rdx  lea -0xc(%rbp),%rax  mov %rax,%rsi  mov %rdx,%rdi  callq 870 <\_ZNSirsERi@plt>  mov -0x10(%rbp),%eax  mov %eax,%esi  lea 0x201381(%rip),%rdi # 202020 <\_ZSt4cout@@GLIBCXX\_3.4>  callq 8d0 <\_ZNSolsEi@plt>  lea 0x1e7(%rip),%rsi # e92 <\_ZStL19piecewise\_construct+0x8e>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov -0xc(%rbp),%eax  mov %eax,%esi  mov %rdx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  lea 0x1cc(%rip),%rsi # e96 <\_ZStL19piecewise\_construct+0x92>  mov %rax,%rdi  callq 890 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rcx  mov -0x10(%rbp),%eax  mov -0xc(%rbp),%esi  cltd  idiv %esi  mov %eax,%esi  mov %rcx,%rdi  callq 8d0 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x2012de(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq 8a0 <\_ZNSolsEPFRSoS\_E@plt>  jmpq a91 <main+0x1e> | if (choice == 3) {  std::cout << num1 << “/“ << num2 << “=” num1 / num2 << std::endl; } |
| mov $0x0,%eax  mov -0x8(%rbp),%rcx  xor %fs:0x28,%rcx  je d1b <main+0x2a8>  callq 8b0 <\_\_stack\_chk\_fail@plt>  leaveq  retq | DisplayMenu();  cin >> choice; |
|  |  |