# CS 410 Project One Proficiency Test Template

## Explain the functionality of the blocks of assembly code.

### “main” function”

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp  lea 0x5eb(%rip),%rsi # 1440 <\_ZStL19piecewise\_construct+0x8>  lea 0x201244(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Start prologue  End prologue  Loads and prints “Hello! Welcome to our Investment Company” |
| callq f6a <\_Z25CheckUserPermissionAccessv>  mov %eax,0x201494(%rip) # 202300 <answer>  mov 0x20148e(%rip),%eax # 202300 <answer>  cmp $0x1,%eax  je e8a <main+0x40>  lea 0x5f2(%rip),%rsi # 1470 <\_ZStL19piecewise\_construct+0x38>  lea 0x20121b(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov 0x201470(%rip),%eax # 202300 <answer>  cmp $0x1,%eax  je e97 <main+0x4d>  jmp e61 <main+0x17> | While loops that calls CheckUserPermissionAccess function.  Compares and authenticates answers for CheckUserPermissionAccess.  If CheckUserPermissionAccess is true, jumps to e8a <main+0x40>.  If CheckUserPermissionAccess is false, output “Please try again.”  While loop to exit loop, else jumps to beginning of the loop to CheckUserPermissionAccess. |
| lea 0x5f6(%rip),%rsi # 1494 <\_ZStL19piecewise\_construct+0x5c>  lea 0x2011fb(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x5ff(%rip),%rsi # 14b0 <\_ZStL19piecewise\_construct+0x78>  lea 0x2011e8(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x614(%rip),%rsi # 14d8 <\_ZStL19piecewise\_construct+0xa0>  lea 0x2011d5(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x625(%rip),%rsi # 14fc <\_ZStL19piecewise\_construct+0xc4>  lea 0x2011c2(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x20140a(%rip),%rsi # 2022f4 <choice>  lea 0x2012cf(%rip),%rdi # 2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq c60 <\_ZNSirsERi@plt>  lea 0x61d(%rip),%rsi # 151a <\_ZStL19piecewise\_construct+0xe2>  lea 0x20119c(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x2013e2(%rip),%eax # 2022f4 <choice>  mov %eax,%esi  mov %rdx,%rdi  callq d00 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x2010aa(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt>  mov 0x2013bd(%rip),%eax # 2022f4 <choice> | While loop  Load and prints menu options of:  What would you like to do?  DISPLAY the client list (enter 1)  CHANGE a client’s choice (enter 2)  EXIT the program (enter 3)  Reads user input and stores it into variable  Loads and prints “You chose” with user input that was entered. |
| cmp $0x1,%eax  jne f43 <main+0xf9>  callq 108b <\_Z11DisplayInfov>  jmp f53 <main+0x109>  mov 0x2013ab(%rip),%eax # 2022f4 <choice> | Compares input to menu option 1  Jumps and calls DisplayInfo Function |
| cmp $0x2,%eax  jne f53 <main+0x109>  callq 1277 <\_Z20ChangeCustomerChoicev>  mov 0x20139b(%rip),%eax # 2022f4 <choice> | Compares input to menu option 2  Jumps and calls ChangeCustomerChoice Function |
| cmp $0x3,%eax  je f63 <main+0x119>  jmpq e97 <main+0x4d>  mov $0x0,%eax  pop %rbp  retq | Compares input to menu option 3  Jumps and sets return value to zero and exits program |
|  |  |

### ChangeCustomerChoice function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp  lea 0x346(%rip),%rsi # 15c8 <\_ZStL19piecewise\_construct+0x190>  lea 0x200e17(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Start prologue  End prologue  Loads and prints “Enter the number of client that you wish to change” |
| lea 0x201063(%rip),%rsi # 2022f8 <changechoice>  lea 0x200f24(%rip),%rdi # 2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq c60 <\_ZNSirsERi@plt> | User inputs client’s number, choosing between 1-5. |
| lea 0x358(%rip),%rsi # 1600 <\_ZStL19piecewise\_construct+0x1c8>  lea 0x200df1(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt> | Loads and prints “Please enter the client’s new service choice (1 = Brokerage, 2 = Retirement)” |
| lea 0x201041(%rip),%rsi # 2022fc <newservice>  lea 0x200efe(%rip),%rdi # 2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq c60 <\_ZNSirsERi@plt> | User inputs client’s new service choice |
| mov 0x20102b(%rip),%eax # 2022f8 <changechoice>  cmp $0x1,%eax  jne 12e0 <\_Z20ChangeCustomerChoicev+0x69>  mov 0x201024(%rip),%eax # 2022fc <newservice>  mov %eax,0x200d82(%rip) # 202060 <num1>  jmp 1342 <\_Z20ChangeCustomerChoicev+0xcb>  mov 0x201012(%rip),%eax # 2022f8 <changechoice> | Compares input to client number 1 and jumps to client’s new service choice. |
| cmp $0x2,%eax  jne 12f9 <\_Z20ChangeCustomerChoicev+0x82>  mov 0x20100b(%rip),%eax # 2022fc <newservice>  mov %eax,0x200d6d(%rip) # 202064 <num2>  jmp 1342 <\_Z20ChangeCustomerChoicev+0xcb>  mov 0x200ff9(%rip),%eax # 2022f8 <changechoice> | Compares input to client number 2 and jumps to client’s new service choice. |
| cmp $0x3,%eax  jne 1312 <\_Z20ChangeCustomerChoicev+0x9b>  mov 0x200ff2(%rip),%eax # 2022fc <newservice>  mov %eax,0x200d58(%rip) # 202068 <num3>  jmp 1342 <\_Z20ChangeCustomerChoicev+0xcb>  mov 0x200fe0(%rip),%eax # 2022f8 <changechoice> | Compares input to client number 3 and jumps to client’s new service choice. |
| cmp $0x4,%eax  jne 132b <\_Z20ChangeCustomerChoicev+0xb4>  mov 0x200fd9(%rip),%eax # 2022fc <newservice>  mov %eax,0x200d43(%rip) # 20206c <num4>  jmp 1342 <\_Z20ChangeCustomerChoicev+0xcb>  mov 0x200fc7(%rip),%eax # 2022f8 <changechoice> | Compares input to client number 4 and jumps to client’s new service choice. |
| cmp $0x5,%eax  jne 1342 <\_Z20ChangeCustomerChoicev+0xcb>  mov 0x200fc0(%rip),%eax # 2022fc <newservice>  mov %eax,0x200d2e(%rip) # 202070 <num5> | Compares input to client number 5 and jumps to client’s new service choice. |
| nop  pop %rbp  retq | Processor does nothing  Removes stored value in %rbp  Returns |
|  |  |

### CheckUserPermissonAccess Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp  push %rbx  sub $0x48,%rsp  mov %fs:0x28,%rax  mov %rax,-0x18(%rbp)  xor %eax,%eax  lea -0x45(%rbp),%rax  mov %rax,%rdi  callq d20 <\_ZNSaIcEC1Ev@plt>  lea -0x45(%rbp),%rdx  lea -0x40(%rbp),%rax  lea 0x588(%rip),%rsi # 1525 <\_ZStL19piecewise\_construct+0xed>  mov %rax,%rdi  callq ce0 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEEC1EPKcRKS3\_@plt>  lea -0x45(%rbp),%rax  mov %rax,%rdi  callq cb0 <\_ZNSaIcED1Ev@plt>  movl $0x0,-0x44(%rbp)  lea 0x567(%rip),%rsi # 1526 <\_ZStL19piecewise\_construct+0xee>  lea 0x2010da(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x20130e(%rip),%rsi # 2022e0 <username>  lea 0x2011e7(%rip),%rdi # 2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq c40 <\_ZStrsIcSt11char\_traitsIcEERSt13basic\_istreamIT\_T0\_ES6\_PS3\_@plt>  lea 0x558(%rip),%rsi # 153d <\_ZStL19piecewise\_construct+0x105>  lea 0x2010b4(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea -0x40(%rbp),%rax  mov %rax,%rsi  lea 0x2011c1(%rip),%rdi # 2021c0 <\_ZSt3cin@@GLIBCXX\_3.4>  callq cd0 <\_ZStrsIcSt11char\_traitsIcESaIcEERSt13basic\_istreamIT\_T0\_ES7\_RNSt7\_\_cxx1112basic\_stringIS4\_S5\_T1\_EE@plt>  lea -0x40(%rbp),%rax  lea 0x545(%rip),%rsi # 1554 <\_ZStL19piecewise\_construct+0x11c>  mov %rax,%rdi  callq c50 <\_ZNKSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEE7compareEPKc@plt>  mov %eax,-0x44(%rbp)  cmpl $0x0,-0x44(%rbp)  jne 1027 <\_Z25CheckUserPermissionAccessv+0xbd>  mov $0x1,%ebx  jmp 102c <\_Z25CheckUserPermissionAccessv+0xc2>  mov $0x2,%ebx  lea -0x40(%rbp),%rax  mov %rax,%rdi  callq c70 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEED1Ev@plt>  mov %ebx,%eax  mov -0x18(%rbp),%rcx  xor %fs:0x28,%rcx  je 1084 <\_Z25CheckUserPermissionAccessv+0x11a>  jmp 107f <\_Z25CheckUserPermissionAccessv+0x115>  mov %rax,%rbx  lea -0x45(%rbp),%rax  mov %rax,%rdi  callq cb0 <\_ZNSaIcED1Ev@plt>  mov %rbx,%rax  mov %rax,%rdi  callq d10 <\_Unwind\_Resume@plt>  mov %rax,%rbx  lea -0x40(%rbp),%rax  mov %rax,%rdi  callq c70 <\_ZNSt7\_\_cxx1112basic\_stringIcSt11char\_traitsIcESaIcEED1Ev@plt>  mov %rbx,%rax  mov %rax,%rdi  callq d10 <\_Unwind\_Resume@plt>  callq cc0 <\_\_stack\_chk\_fail@plt>  add $0x48,%rsp  pop %rbx  pop %rbp  retq | Start prologue  End prologue  Initialize string password  Display prompts, gets user input and compares password  Prints enter username  Inputs username  Prints enters password  Inputs password  Compares password  IF password is incorrect, return 2  IF password is correct, return 1 |
|  |  |

### DisplayInfo Function

| **Assembly Code Block** | **Explanation of Functionality** |
| --- | --- |
| push %rbp  mov %rsp,%rbp  lea 0x4c2(%rip),%rsi # 1558 <\_ZStL19piecewise\_construct+0x120>  lea 0x201003(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x200f24(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt> | Start prologue  End prologue  Display output header “Client's Name | Service Selected (1 = Brokerage, 2 = Retirement)” |
| lea 0x4de(%rip),%rsi # 159c <\_ZStL19piecewise\_construct+0x164>  lea 0x200fdb(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x200f3f(%rip),%rsi # 202010 <name1>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x4c0(%rip),%rsi # 15a0 <\_ZStL19piecewise\_construct+0x168>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x200f6f(%rip),%eax # 202060 <num1>  mov %eax,%esi  mov %rdx,%rdi  callq d00 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x200ecb(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt> | Display output for first client  “Bob Jones” with selected option |
| lea 0x49b(%rip),%rsi # 15b2 <\_ZStL19piecewise\_construct+0x17a>  lea 0x200f82(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x200ef6(%rip),%rsi # 202020 <name2>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x467(%rip),%rsi # 15a0 <\_ZStL19piecewise\_construct+0x168>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x200f1a(%rip),%eax # 202064 <num2>  mov %eax,%esi  mov %rdx,%rdi  mov %rax,%rdx  mov 0x200e72(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt> | Display output for second client  “Sarah Davis” with selected option |
| lea 0x446(%rip),%rsi # 15b6 <\_ZStL19piecewise\_construct+0x17e>  lea 0x200f29(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x200ead(%rip),%rsi # 202030 <name3>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x40e(%rip),%rsi # 15a0 <\_ZStL19piecewise\_construct+0x168>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x200ec5(%rip),%eax # 202068 <num3>  mov %eax,%esi  mov %rdx,%rdi  callq d00 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x200e19(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt> | Display output for third client  “Amy Friendly” with selected option |
| lea 0x3f1(%rip),%rsi # 15ba <\_ZStL19piecewise\_construct+0x182>  lea 0x200ed0(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x200e64(%rip),%rsi # 202040 <name4>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x3b5(%rip),%rsi # 15a0 <\_ZStL19piecewise\_construct+0x168>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x200e70(%rip),%eax # 20206c <num4>  mov %eax,%esi  mov %rdx,%rdi  callq d00 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x200dc0(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt> | Display output for fourth client  “Johnny Smith” with selected option |
| lea 0x39c(%rip),%rsi # 15be <\_ZStL19piecewise\_construct+0x186>  lea 0x200e77(%rip),%rdi # 2020a0 <\_ZSt4cout@@GLIBCXX\_3.4>  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x200e1b(%rip),%rsi # 202050 <name5>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  lea 0x35c(%rip),%rsi # 15a0 <\_ZStL19piecewise\_construct+0x168>  mov %rax,%rdi  callq c90 <\_ZStlsISt11char\_traitsIcEERSt13basic\_ostreamIcT\_ES5\_PKc@plt>  mov %rax,%rdx  mov 0x200e1b(%rip),%eax # 202070 <num5>  mov %eax,%esi  mov %rdx,%rdi  callq d00 <\_ZNSolsEi@plt>  mov %rax,%rdx  mov 0x200d67(%rip),%rax # 201fd0 <\_ZSt4endlIcSt11char\_traitsIcEERSt13basic\_ostreamIT\_T0\_ES6\_@GLIBCXX\_3.4>  mov %rax,%rsi  mov %rdx,%rdi  callq ca0 <\_ZNSolsEPFRSoS\_E@plt> | Display output for fifth client  “Carol Spears” with selected option |
| nop  pop %rbp  retq | Processor does nothing  Removes stored value in %rbp  Returns |
|  |  |