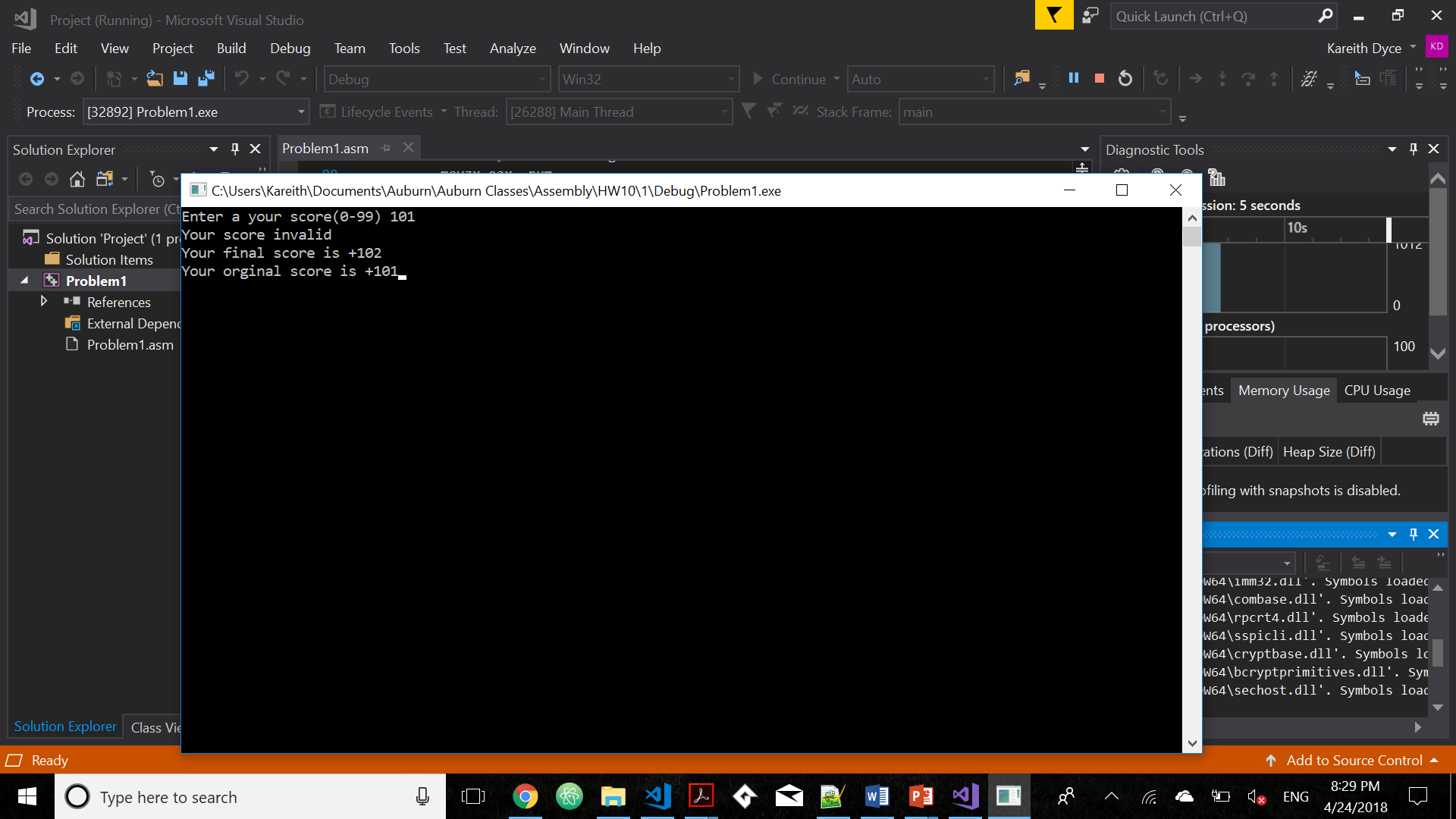
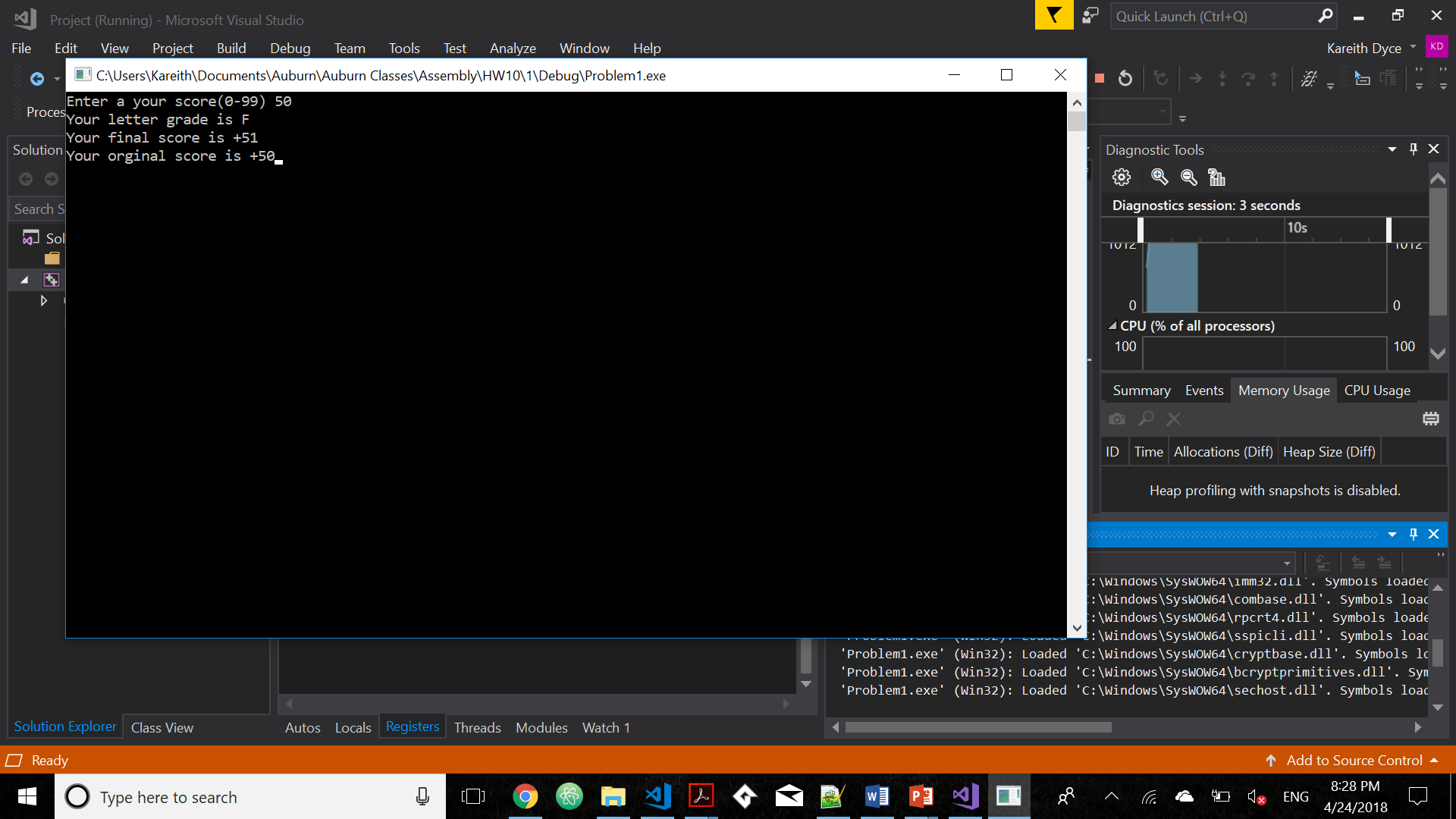
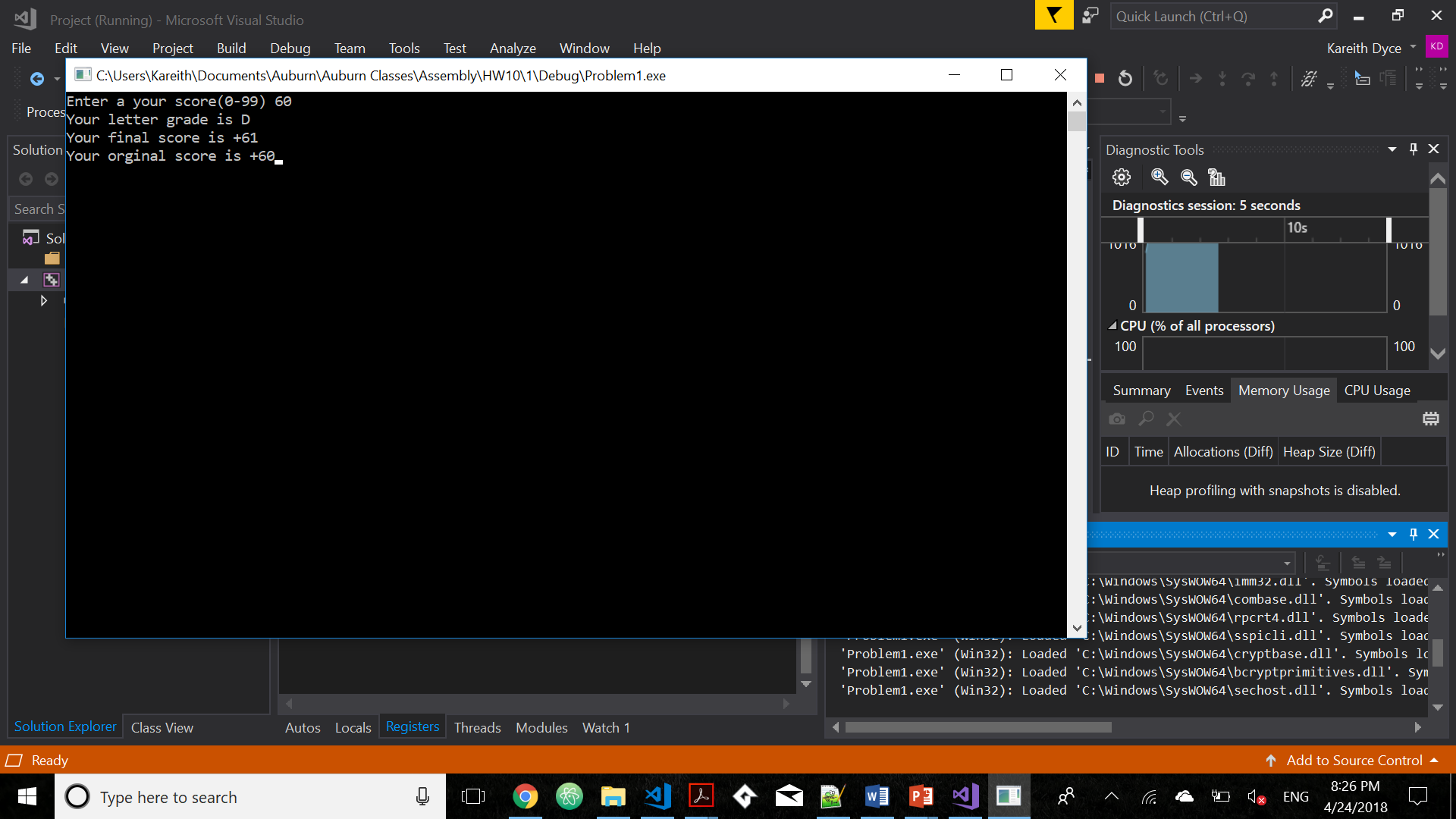
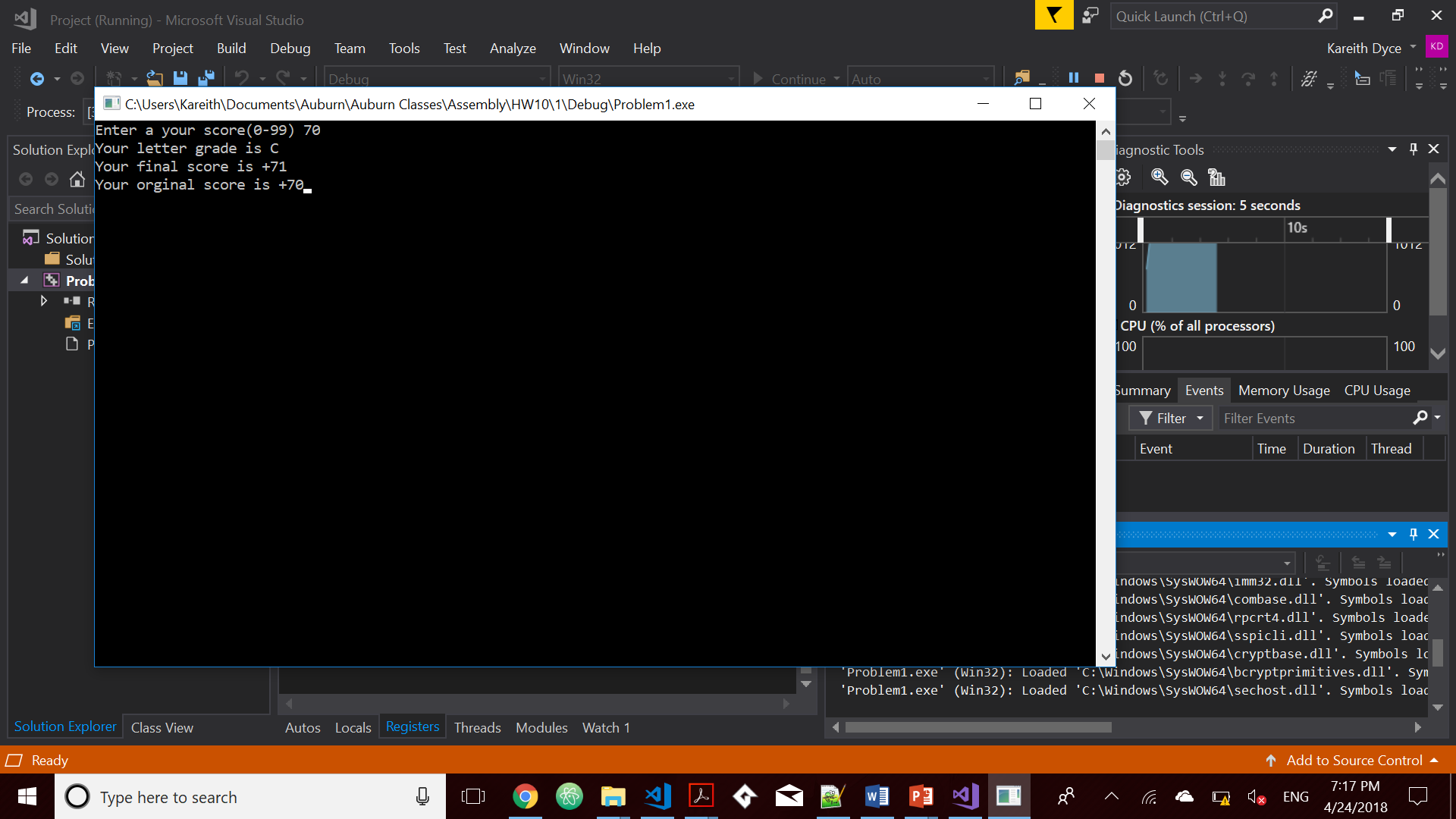
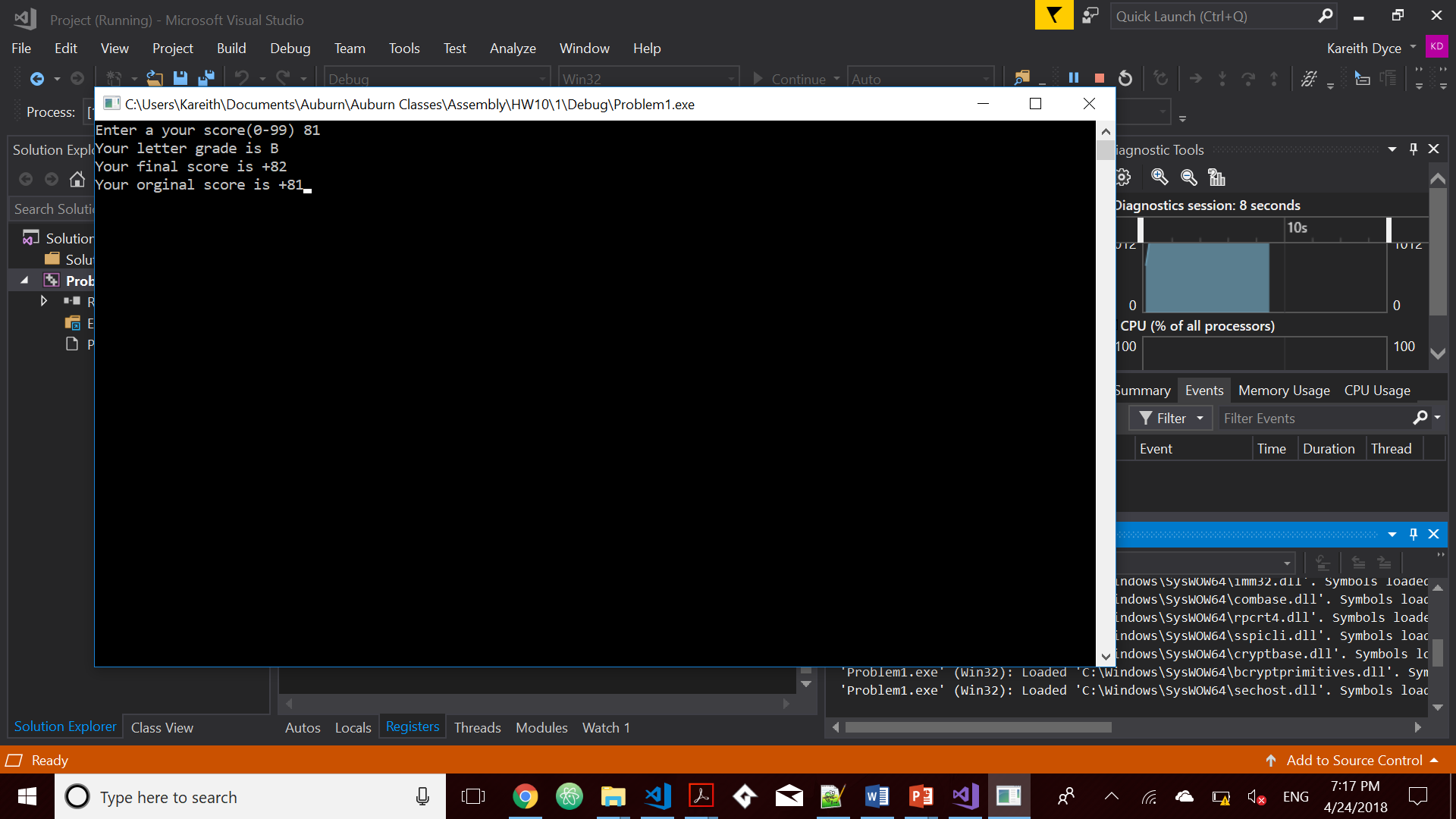
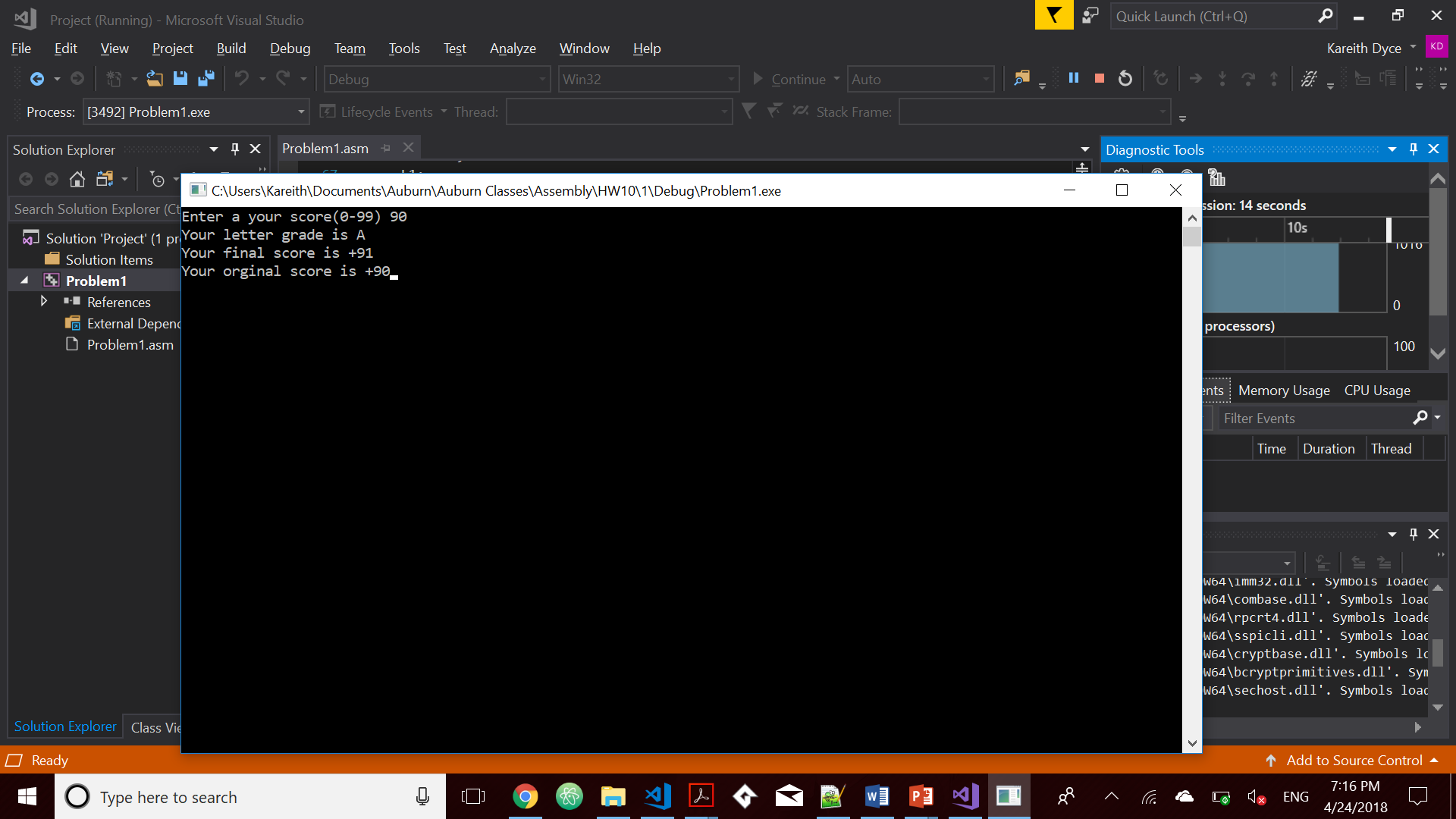
# Comp 3350: Computer Organization & Assembly Language

# HW # 10: Theme: Strings and Arrays

*(All main questions carry equal weight. Credit awarded to only those answers for which work has been shown.)*

1. [Case Table] Write a program that asks the user to enter a score. It then adds 1 to the score, creating a final score and prints the letter grade based on the final score (see table below). The program should display your original score, final score as well as the letter grade. You should reference the section of the text that discusses Table Driven Selection. Use the following data as a guide for letter grade and score range association:

|  |  |
| --- | --- |
| **Score Range** | **Letter Grade** |
| 89 – 100 | A |
| 79 – 88 | B |
| 69 - 78 | C |
| 59 – 68 | D |
| 0 - 58 | F |



Please embed your code into your homework submission along with a screen shot post execution.

*Include Irvine32.inc*

*.386*

*.stack 4096*

*ExitProcess proto,dwExitCode:dword*

*.data*

*CaseTable WORD 89*

*DWORD Process\_A*

*EntrySize = ($ - CaseTable)*

*WORD 79*

*DWORD Process\_B*

*WORD 69*

*DWORD Process\_C*

*WORD 59*

*DWORD Process\_D*

*WORD 0*

*DWORD Process\_F*

*NumberOfEntries = ($ - CaseTable) / EntrySize*

*msgA BYTE "Your letter grade is A", 0*

*msgB BYTE "Your letter grade is B", 0*

*msgC BYTE "Your letter grade is C", 0*

*msgD BYTE "Your letter grade is D", 0*

*msgF BYTE "Your letter grade is F", 0*

*string1 Byte "Enter a your score(0-99) ", 0*

*string2 Byte "Your orginal score is ", 0*

*string3 Byte "Your final score is ", 0*

*string4 Byte "Your score invalid ", 0*

*num WORD ?*

*.code*

*Process\_A PROC*

*mov edx,OFFSET msgA*

*ret*

*Process\_A ENDP*

*Process\_B PROC*

*mov edx,OFFSET msgB*

*ret*

*Process\_B ENDP*

*Process\_C PROC*

*mov edx,OFFSET msgC*

*ret*

*Process\_C ENDP*

*Process\_D PROC*

*mov edx,OFFSET msgD*

*ret*

*Process\_D ENDP*

*Process\_F PROC*

*mov edx,OFFSET msgF*

*ret*

*Process\_F ENDP*

*main PROC*

*mov edx, offset string1*

*Call WriteString*

*Call ReadInt*

*mov num, ax*

*inc eax*

*cmp ax, 100*

*ja L3*

*mov ebx, OFFSET CaseTable*

*mov ecx, NumberOfEntries*

*L1:*

*cmp ax, [ebx]*

*jbe L2 ;*

*call NEAR PTR [ebx + 2]*

*call WriteString*

*call Crlf*

*jmp L4 ; exit the search*

*L2:*

*add ebx,EntrySize*

*loop L1*

*L3:*

*mov edx, OFFSET string4*

*call writestring*

*call Crlf*

*L4:*

*mov edx, offset string3*

*call Writestring*

*call Writeint*

*call Crlf*

*mov edx, offset string2*

*movzx eax, num*

*call Writestring*

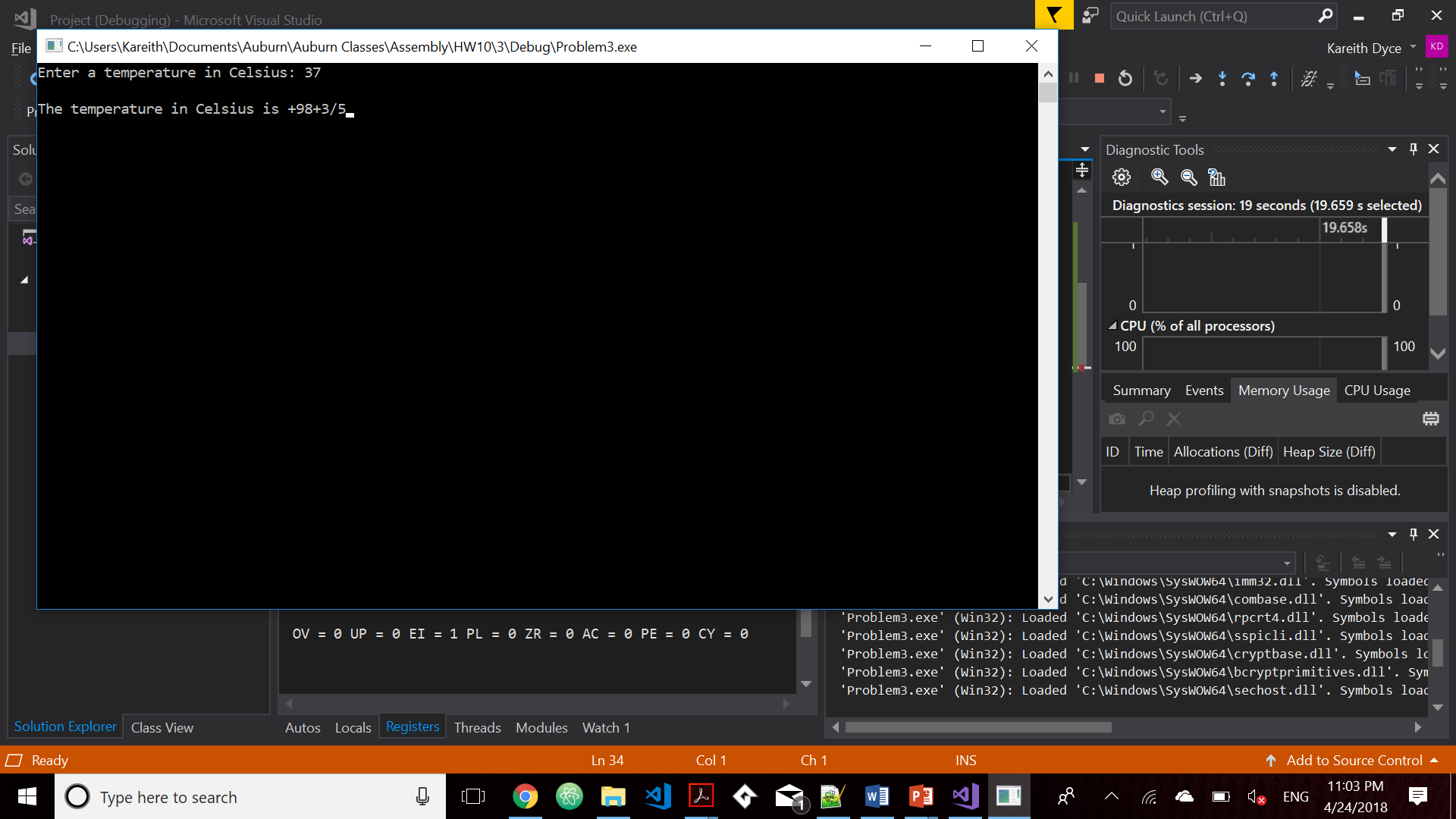
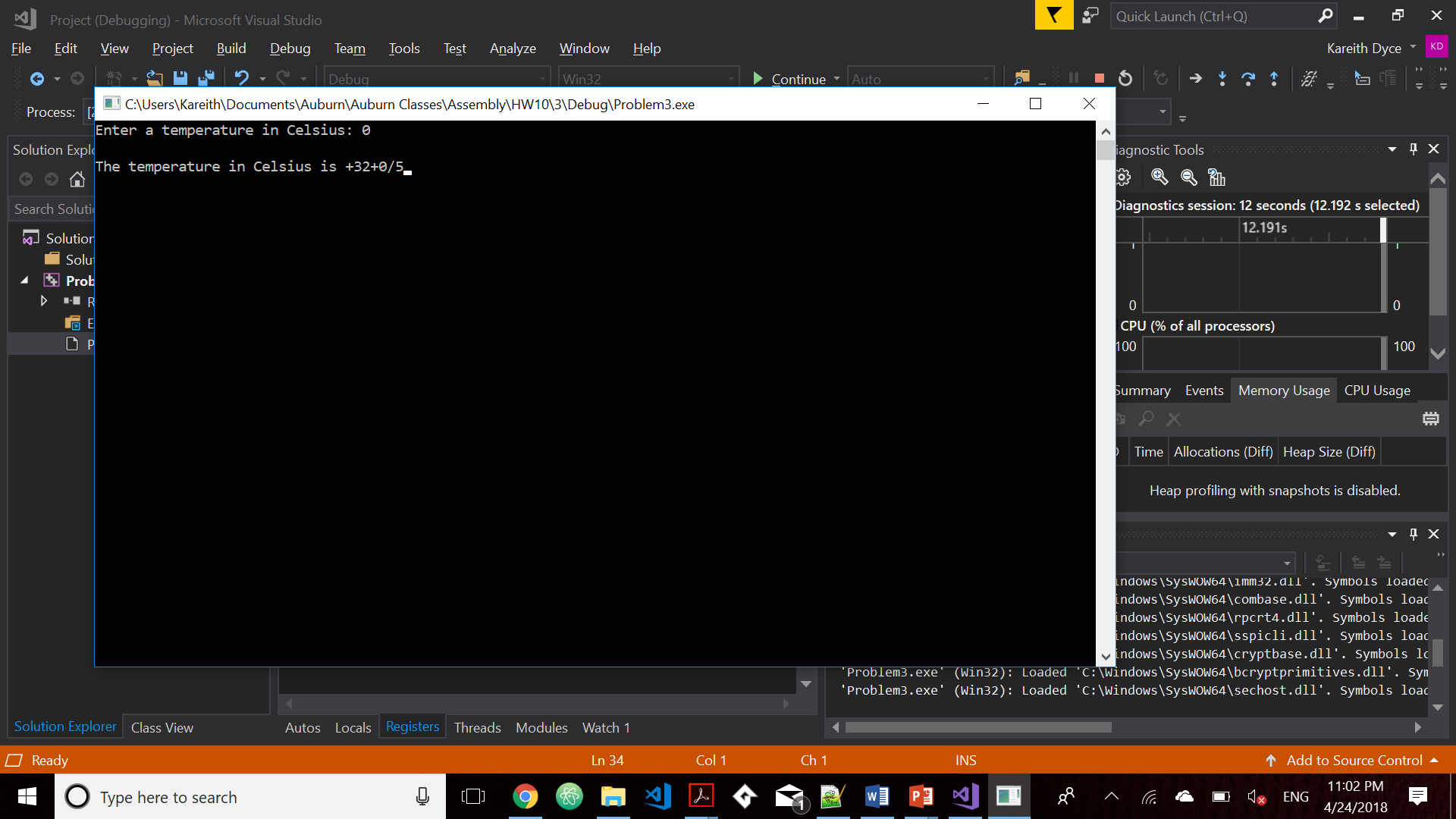
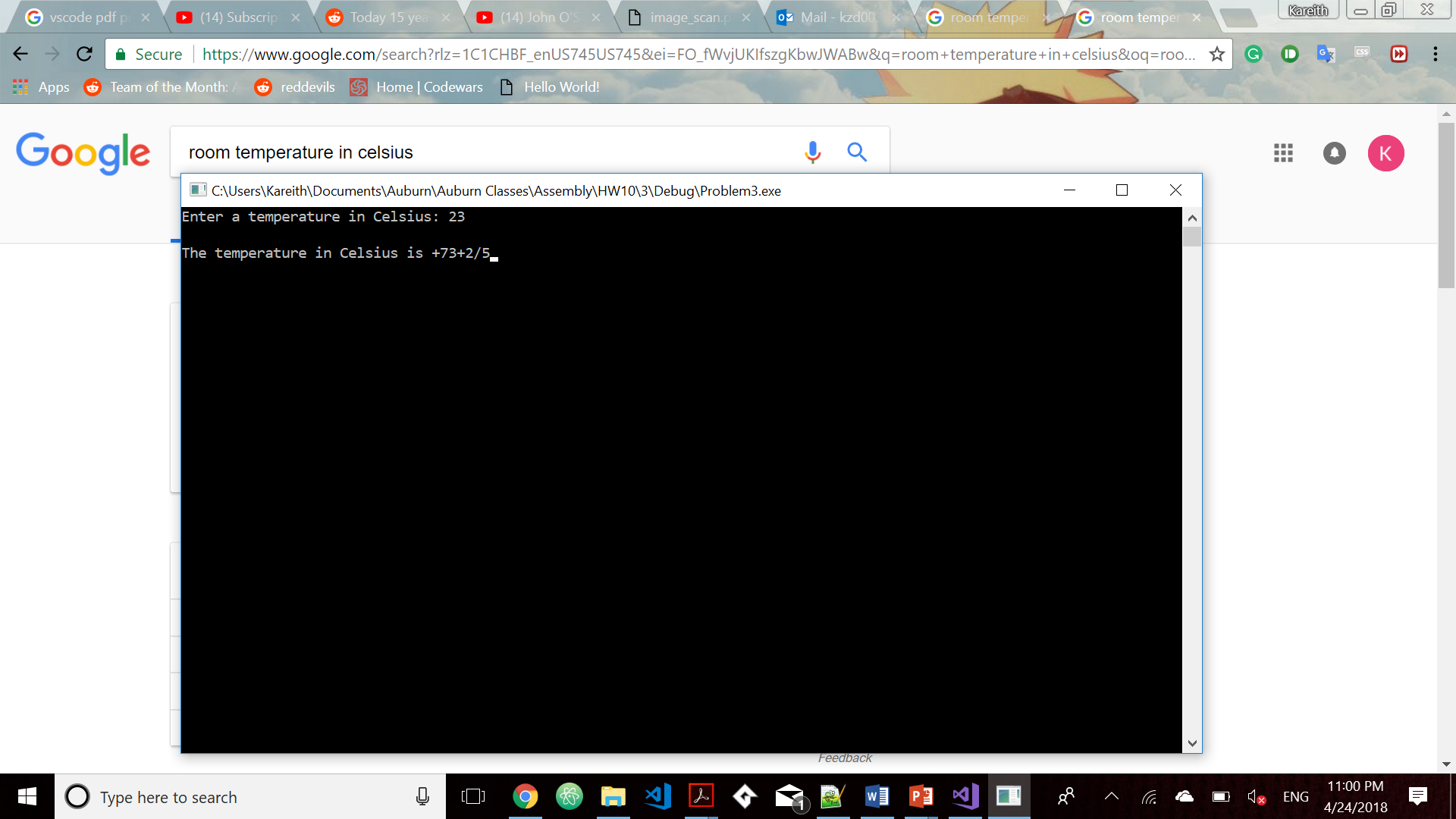
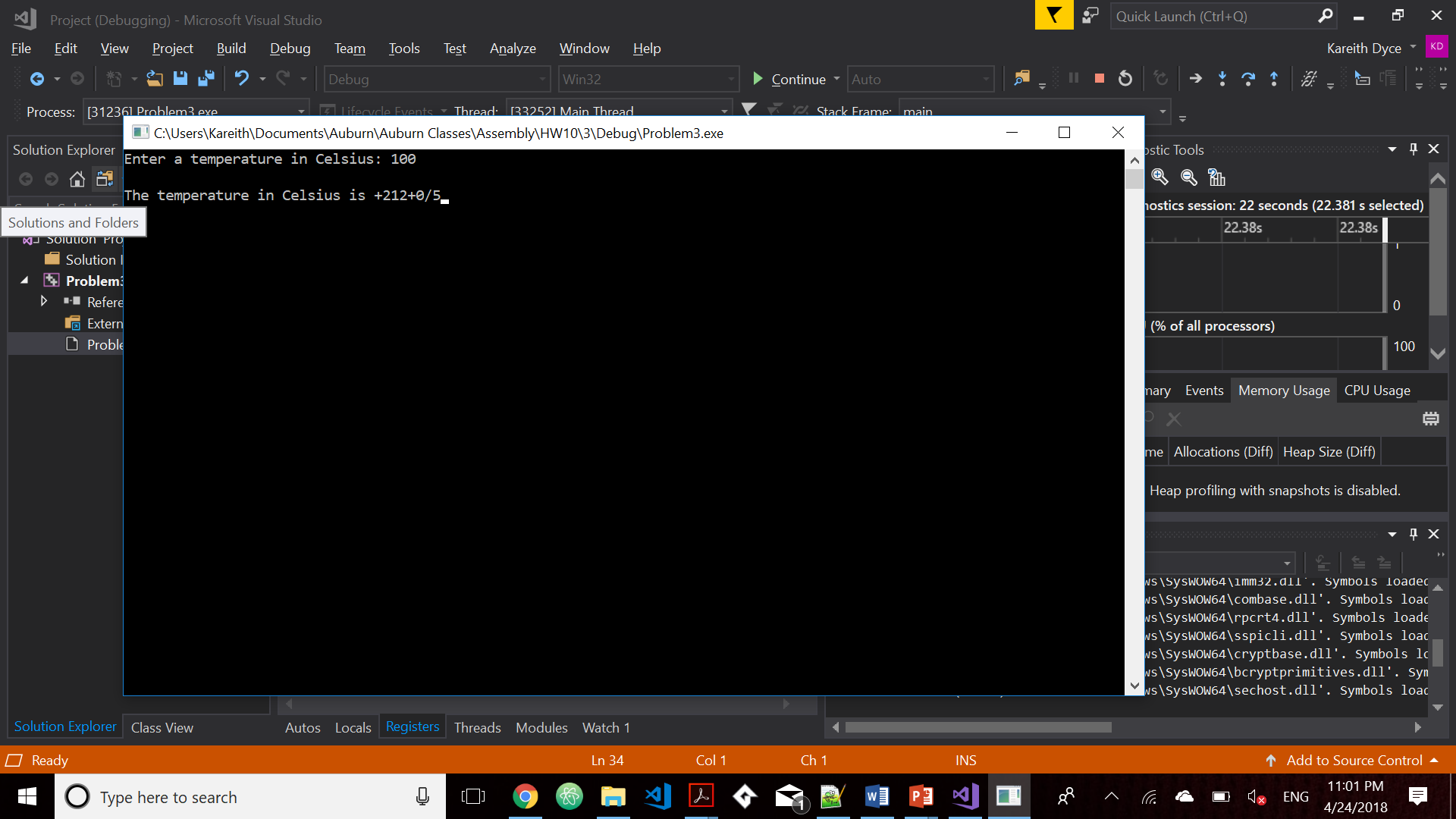
*call Writeint*

*invoke ExitProcess,0*

*main endp*

*end main*

1. [Strings] Write a program that copies bytes from source to target. You must use string instructions to accomplish the job. Declare the source and target locations in the data segment. The source string should be your name.
2. [General Programming] Write a program that converts the temperaturein Celsius to *F* in Farenheit using *F = C\*9/5 + 32.* For ease of programming you can display the result in fractions, i.e. *20 1/9* (no need to use floats, just display the quotient, the slash character and the digit 9). Show the runs for freezing, boiling point, room temperature and human body temperature. Provide screen shots of the runs along with your program.



1. [Strings] Write a program that searches for a character in a string. You should set the EDI pointer to point to the character found. Test the program thoroughly using various strings, including your name. Provide screen shots of the runs along with your program. You must use string instructions in your program.

