**Python Lab 3a: if statements with else and elif**

|  |  |
| --- | --- |
| 1. Write an if statement to determine if a number is divisible by 5. | Ask the user to enter a number. Determine if the number is divisible by 5.  **if (number % 5 == 0):**  **print (str(number) + ' is divisible by 5')** |
| 2. Write an if/else statement. | An else statement can follow an if statement. The code in an else statement is executed if the expression in the if statement is false.  **if (number % 5 == 0):**  **print (str(number) + ' is divisible by 5')**  **else:**  **print (str(number) + ' is divisible by 5')**  Notice that an else statement does not have a Boolean expression.  Write an additional if/else statement to see if the number is divisible by 2.  Test your code with many different inputs. |
| 3. Write an if/elif/else sequence. | The command elif can be used to make a series of if statements more efficient. As soon as one of the expressions is true, the other ones are skipped. elif is short for 'else if'.  Ask the user to enter a State/Province name, and print out the capital of that state. Your code should handle at least 6 different inputs. Use an if/elif/else structure  **if (state == 'Wisconsin'):**  **print ('Madison')**  **elif (state == 'Colorado)':**  **print('Denver')**  **…**  **…**  **else:**  **print ('I do not know that one')**  Test your code with many different input values. |
| 4. Use elif in a function definition. | At a certain public pool, entrance prices are as follows:  Under 2 years: free, Age 2–11: $3, Age 11–60: $6, Over 60: $4  Complete a function that takes the age in years as input and returns the price in dollars (without the dollar sign).  **def pool\_admission(age):**  **if (age < 2):**  **return 0**  **elif (age < 12):**  **return 3**  Notice that elif reduces possibilities and makes the code easier to read. |

**Python Lab 3b: Translating the Vole Operations into words**

**Before doing this lab, turn to Appendix C in the text.**

|  |  |
| --- | --- |
| 1. Define a function which takes as input a vole command such as 0x3, and returns the corresponding string as defined in Appendix C. Enter and test the following code. Note that we use different names for the command in our main program and the op\_code in the definition. | **# command is the hex-digit Vole command**  **command = 0x3**  **def vole\_description(op\_code):**  **if (op\_code == 0x1):**  **return 'LOAD memory '**  **elif (op\_code == 0x2):**  **return 'LOAD the value'**  **# more code goes here**  **else:**  **return 'bad op\_code'**  **print (vole\_description(command))** |
| 2. Use Appendix C to add the remaining op-codes to your program. | Test your program with many different commands. |
| 3. A full Vole command has 4 hex digits. It's possible to use integer division to isolate the first hex digit. To understand how, consider a base-10 integer such as 8723. To isolate the first digit, we can perform integer division by 1000.  **thousands\_digit = 8723 // 1000**  **# stores 8** | Make the following changes to your program.  **# full 4-hex-digit Vole command**  **full\_command = 0x75a4**  **# operation is the left-most hex digit**  **operation = full\_command // 0x1000**  **print (hex(operation))**  **def vole\_description(op\_code):**  **….**  **print (vole\_description(operation))** |
| 4. To get the other digits, we will use the following commands:  **#bit\_pattern is the last two hex-digits**  **bit\_pattern = full\_command % 0x100**  **print (hex(bit\_pattern))**  **#register is the second hex-digit**  **register = full\_command // 0x100 % 0x10**  **print (hex(register))** | Enter the code to the left. Run it. Then modify vole\_description to write complete descriptions that match those in Appendix C. You will pass three values into the function, as shown below:  **def vole\_description(op\_code, reg, value):**  **if (op\_code == 0x1):**  **# put the following on a single line**  **return 'LOAD register ' + reg + ' with the contents of memory cell ' + value**  **print (vole\_description(operation, hex(register), hex(bit\_pattern)))** |

Test your program by entering examples from Section 2.2.