Simon Fraser University
Computing Science 120
PRACTICE Midterm Examination

Time: X minutes

Last/Family Name (please, print): ________________________________

First Name (please, print): ________________________________

Student Number: ________________________________

Signature: ________________________________

Instructor: Anne Lavergne

This examination has X pages inclusive.

Verify that you have a complete paper.

Instructions:

✓ One cheat sheet allowed. This cheat sheet must …
  • be 8 ½ by 11 inches and one-sided,
  • be handwritten,
  • have your name and student number on its top right corner, and
  • be handed in with this midterm examination.
✓ No books, calculators, computers, cell phones, or other materials may be used.
✓ Read each question carefully before answering it.
✓ You must use Python 3. Always comment your code and always provide a useful user interaction.
✓ You can assume that all Python code fragments given in this exam are syntactically correct, unless stated.
✓ The marks for each question are given in [ ]. Use this to manage your time.

Good luck!
Please, answer on this exam paper.

1. [2 marks] Write the header of a Python function which, given 3 numbers, figures out the smallest number. The 3 numbers are not entered by the user within the function.
   Warning: Marks will be deducted if you write more than what the question is asking for.
2. a) [6 marks] Problem Statement: Write a **Python function** that takes a string of float values as a parameter and returns a float value. The float values contained in the string represent prices and the returned float value represents the total of these prices.

For example, if the string is "2.35 0.75 18.10 3.75", then the function returns **24.95** and if the string is "1.75 34.97 33.10", the function returns **69.82**. Note that the prices are separated by one white space character.

You can assume that the strings passed to your function will always be composed of prices as exemplified above or be empty.

If you find it useful to start by designing an algorithm, feel free to do so.
b) [3 marks] List the test data you would use to completely test the function you created in 2. a) (on the previous page) and list the corresponding results you would be expecting.
3. [4 marks] Consider the following Python code fragment:

```python
def spell(theWord):
    theLength = len(theWord)
    index = 1
    for letter in theWord:
        print("Letter %i is: %s" % (index, letter))
        index = index + 1
    return theLength

# Main
aWord = input("Please, enter a word: ")
print("The word you entered is %s" %aWord)
aLength = spell(aWord)
print("This word has %i letters." %aLength)
```

Write what this Python code fragment produces when it is executed and the user enters 264. If the Python code fragment produces an error, write “Error” and explain what causes the error.
4. [10 marks] **Problem Statement:** Write a complete Python program that contains one function called `replaceEven(...)`, which requires a number `num` and a character `symbol`. With these, the function builds a string by replacing the even integers (from 1 to `num`, inclusively) with the character `symbol`.

For example, if `num = 9` and `symbol = “^”`, the `replaceEven(...)` function creates and prints the string "1^3^5^7^9", and if `num = 12` and `symbol = “!”`, the `replaceEven(...)` function creates and prints the string "1!3!5!7!9!11!".

Once the function `replaceEven(...)` has created and printed a string, it returns the number of time it replaced an even number with the character `symbol`. The Main section of your program then prints this number.

If you find it useful to start by designing an algorithm, feel free to do so.