NAME: _____ RPI ID _____

CS1010 Introduction to Computer Programming Spring 2019 Exam 1

Please read the following pledge, then sign and print your name on the spaces provided, certifying the statement:
On my honor as a Rensselaer Polytechnic Institute student, I have abided by academic integrity standards on this exam, which means that I will not give or take answers from anyone.

_____

Your Signature and Date

_____

Your PRINTED name

Rules: There are ***5 questions*** in all to be completed in ***1 hour 50 minutes***.

1. Work entirely alone. Do not give or solicit assistance from any other student. Academic dishonesty will not be tolerated.
2. Sit in your assigned seat.
3. Turn off cell phones and smart phones.
4. The exam allows use of hand written notes (2 pages A4 size) for reference.
5. Feel free to use the restrooms as necessary. Just leave all your materials at your seat.
6. If you have a question, bring it down to the front so as to minimize disruption.

Question 1_____

Question 2_____

Question 3_____

Question 4_____

Question 5_____

Total (From 100 points): _____

Question 1. Write in the appropriate expression, term or phrase (**One or two words only**).

[**20 points: 2 points each**]

1. Where do running programs live in your computer?

2. What is the brain of your computer?

3. What is the minimum storage unit in a computer called?

4. Program written in a high level language is called?

5. Name any one category of programming languages?

6. What object type in Python describes Decimal Numbers?

7. What sign is used for equality comparison in Python?

8. What object types take only True or False values in Python?

9. A Python Program is called a:

10. What built-in function in Python prints values?

Q2. a) Given two int values, return their sum. Unless the two values are the same, then return double their sum. (10 points).

Test cases:
sum_double(1, 2) → 3
sum_double(3, 2) → 5
sum_double(2, 2) → 8

Solution:

b) Given 2 int values and a parameter called negative (Boolean), return True if one number is negative and one is positive. Except if the parameter "negative" is True, then return True only if both are negative. (20 points).
Test Cases:
pos_neg(1, -1, False) → True
pos_neg(-1, 1, False) → True
pos_neg(-4, -5, True) → True

Solution:

Q3. What will be the Boolean output of the following operations (**10 points: 2 points each**)

a. 7<=0

b. (3+5)==8

c. (not(x < 15 and y >= 3)) == (not x>=15)or(not y<3)

d. If x = True, y = True, then what is the outcome of (x and True == False)

e. 5!=5

Q4. What is wrong with the following code. Assume each of the following is a separate program. Find the first error in the code that prevents it from generating output. If there is an error describe it in the solutions box on the right. If there is no error simply write NO ERROR. (**20 points: 5 points each**)

Code	Solution

import math
def f1(v,x,y,z):
  p=2v+x+y+z
  p=math.sqrt(p)
  return p
print("square root is",f1(3,3,3,3))	

def diff_true(a, b):
 diff = a - b
 if diff >= 0:
  return True
 return False	

def isnotequal(a,b): return (a =! b)	

k = 20
if (k == 10):
  # First if statement
if (k < 15):
  print ("k is is in first if statement")
  if (k < 12):
  print ("k is in the nested if")
  else:
  print ("k is in else block of nested if")	

Q5.

a. **String Operation:** Given a string ‘a’, a = ‘Welcome to New York!’

Write a single line of code to accomplish the following: (**10 points: 2 points each**)

Question	Solution/code
Return the string that says ‘New York!’	
Return the string that returns ‘New York!’ in all capital letters	
Return a new string that uses only the first, fifth, eighteenth and ninetheenth characters. Repeat the resulting string 3 times. For a this should give: ‘WorkWorkWork’	
Return the string a with all lowercase letters.	
Return the string a such that all lower case ‘o’ are replaced by 0 (zero) to get: 'Welc0me t0 New Y0rk!'	

b. What is the output of the following code .There is no syntax error here. (**10 points: 5 points each)**

Question	Output
def f(a): return a[::-1] f('google')	

def make_tags(tag, word):
  st1='<'+tag+'>'
  st2='<'+'/'+tag+'>'
  return st1+word+st2
make_tags('address', 'here')	