NAME: \_\_\_\_Timothy Samec\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ RPI ID \_\_661759192\_\_\_\_\_\_\_\_

**CS1010 Introduction to Computer Programming Spring 2019 Final Exam**

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 ***10 questions*** in all to be completed in ***3 hours***.

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 notes (3 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.

Total (From 100 points): \_\_\_\_\_\_\_\_\_\_43\_\_\_\_\_\_\_

**Question 1.** Choose the correct answer (out of the four options). Circle the option that you select. **(20 points – 2 points each)**:

1. What part is called the brain of the computer:
2. Random Access Memory (RAM)
3. Central Processing Unit (CPU)
4. Software
5. Hardware
6. The error encountered when dividing a finite number by zero:
7. Index Error
8. Zero Division Error
9. Type Error
10. Invalid Syntax -2
11. Which step is NOT included in structured programming:
12. Problem divided into smaller sub-problems
13. Each sub-problem is analyzed
14. A complex problem is solved as a whole/single problem
15. A solution for a sub-problem is obtained
16. Which list comprehension returns odd numbers in the range 0 to 8:
17. [x for x in range(8) if x % 2 != 0]
18. [x for x in range(8) if x % 2 == 0]-2
19. [x\*\*3 for x in range(0,8)]
20. [x\*\*2 for x in range(8)]
21. Which of the following does not create a Python Set:
22. {1+2, 3, "a"}
23. {3, 3, "a"}
24. set([1,2,2,2,2,3])
25. { }
26. Profiling a Program means:
27. Measuring the time and memory a program uses
28. Finding characteristics of a program
29. Categorizing a program-2
30. Not executing a program
31. If x=3, y= 5 then x==y returns:
32. True
33. None
34. False
35. Error
36. What is the name of the object that represents ordered sequence of characters:?
37. Sets
38. Strings
39. Dictionary
40. Tuples
41. If s = ‘House’, then s[-1] will return:
42. H
43. O
44. S-2
45. E
46. Which one of the following is a built-in function in Python:
47. print()
48. function()
49. Module
50. typecast

**Question 2.** Explain what each line/lines of code does. **(10 points: 1 point each)**

|  |  |
| --- | --- |
| **Code** | **Output** |
| test = {} | Makes a set called ‘test’ -1 |
| d = d1.copy() | Make a copy of d1 under the new term d |
| s = 'Hello World'  s[len(s)-1] | Taking the length of s and subtracting it by 1  -1 |
| test\_list=[1.6,2.5,3.8,4.1]  sum(test\_list) | Summing the list |
| ls1 = [5,1,2,4,3]  ls1.append('add me to list') | Changing the list to add ‘add me to list’ at the end |
| {2,1,3} | A set with 2,1,3 in it |
| {1,2,3} | {2,3,4} | -1 |
| people.intersection(vampires) | Finding the middle of the word-1 |
| s={1,2,3,4,5}  s.pop() | Removing a random element from the set |
| tup1=('a','b','a')  tup1.count('a') | Counting how many ‘a’ there are in tup1 |

**Question 3.** In the following table there is some code given and a statement (in bold) follows the code. **For the statement**, answer each part in True or False: (**10 points: 2 points each**)

|  |  |
| --- | --- |
| **Code** | **Answer** |
| def max\_of\_two( x, y ):  if x > y:  return x  return y  max\_of\_two (12,14)  **The code/function above returns 12.** | True-2 |
| z=open('test.txt','r')  data = z.read()  z.close()  **The code above reads a text file and saves the contents in a variable called ‘data’.** | True |
| def isright(a,b,c):  return (a^2+b^2 ==c^2)  isright(2,3,4)  **The code/function above returns True** | False |
| def common\_end(a, b):  if len(a)>0 and len(b)>0:  if a[0]==b[0] or a[-1]==b[-1]:  return True  return False  return False  common\_end([1, 2, 3], [7, 3])  **The code/function above returns False** | False |
| def sublist(l1,l2):  if l2[0] in l1:  return True  return False  sublist([1,2,3,4,5],[2,3])  **The code/function above returns True** | False-2 |

**Question 4**. 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. (**15 points: 3 points each**)

|  |  |
| --- | --- |
| Code | Solution |
| def max\_end3(nums):  newlist=[nums[0],nums[-1]]  newelement=max(newlist)  l=[newelement]\*len(nums)  return l | The return function will not output anything-3 |
| a = [1, 2, 3]  print ("Element = %d" %(a[3])) | There is no ‘d’ like stated in the print() -3 |
| ﻿my\_dict = {'data1':100,'data2':-54,'data3':247}  result=1  for key,value in my\_dict:  result=result \* my\_dict[key]  print(result) | There is no ‘key’ defined so there is nothing to input/output here-3 |
| ﻿s1=input("Enter first string:")  s2=input("Enter second string:")  a=list(set(s1)|set(s2))  print("The letters are: ",a) | There are not integers In the ‘sets’ so its impossible to make a list-3 |
| E= {1,2,3,9,10,11,12}  F= {4,5,6,8,9,10,11,12,13}  len(E||F) | No Error-3 |

**Question 5**.

**Dictionary Operations:**

Write a **single line (or at most 2 lines)** of code to accomplish the task stated in Bold: (**10 points: 2 points each**)

|  |  |
| --- | --- |
| Question | Solution/code |
| my\_dict = {'key1':30,'key2':[12,23,33],'key3':['val1','val2','val3']}  **Print/output ‘val3’** | Print(my\_dict)  if ‘val3’ in my\_dict:  print(my\_dict[‘val3’]-2 |
| Dict = { 1 : 'Welcome', 2 : 'To', 3 : 'NY',  'X' : {4 : 'NY', 5 : 'Is', 6 : 'Great'},  'Y' : {7 : 'Empire', 8 : 'State'}}  **Remove the key (=1) that has value ‘Welcome’** | del Dict[‘Welcome’]  Dict-2 |
| m1 = {'a': 10, 'b': 20}  m2 = {'x': 30, 'y': 20}  **Concatenate the above two dictionaries.** | m1.update(m2)  print(m1) |
| dictionary={'one':5, 'two':1, 'three':6, 'four':10}  **Remove all elements of the dictionary given.** | dictionary.clear() |
| Bag={ }  **Add a key called money with value 100 to Bag.** | ﻿  Bag[money] = 100 |

**Question 6**

Write a Python Program to find the index of the two largest integers in a given list L. Write your algorithm using pseudo-code or flowchart first (3 points) and then write the code (2 points). (**5 points)**

L= [80,84,47,48,37,12,96,10,32,46]

**Algorithm:**

1. Take the given list
2. Sort the list from smallest to largest
3. Take the last digit because it is the largest
4. Take the second to last digit because it is the second largest
5. Index the last two numbers
6. Output/Return the last 2 numbers-3

**Code:**

L = [80,84,47,48,37,12,96,10,32,46]

Def find\_two\_largest(L):

Temp\_list = L[:]

Temp\_list.sort()

Largest = Temp\_list[:-1]

Next\_Largest = Temp\_list[:-2]

Max1= L.index(Largest)

Max2= L.index(Next\_Largest)

Return(Max1, Max2)

output: (96,84)

**Question 7:** Select the output of each code from the options given. Circle the option that you select. (**10 points: 2 points each**)

|  |  |  |
| --- | --- | --- |
| **No.** | **Question/Code** | **Options/Answer** |
| **1** | def f(value, values):  v = 1  values[0] = 44  t = 3  v = [1, 2, 3]  f(t, v)  print(t, v[0]) | **A.** 1 44 **B.** 3 1 **C.** 3 44 **D.** 1 1 |
| **2** | def f(i, values = []):  values.append(i)  print (values)  return values  f(1)  f(2)  f(3) | **A.** [1] [2] [3] **B.** [1, 2, 3] **C.** [1] [1, 2] [1, 2, 3] **D.** 1 2 3 |
| **3** | dict = {'c': 97, 'a': 96, 'b': 98}  for x in sorted(dict):  print (dict[x]) | **A.** 96 98 97 **B.** 96 97 98 **C.** 98 97 96 **D.** 98 98 98 |
| **4** | box = {}  jars = {}  crates = {}  box['biscuit'] = 1  box['cake'] = 3  jars['jam'] = 4  crates['box'] = box  crates['jars'] = jars  print (len(crates[box])) | **A.** 1 **B.** 3 **C.** 4 **D.** Type Error |
| **5** | my\_dict = {}  my\_dict[(1,2,4)] = 8  my\_dict[(4,2,1)] = 10  my\_dict[(1,2)] = 12  sum = 0  for k in my\_dict:  sum += my\_dict[k]  print (sum)  print(my\_dict) | **A.** 70 **B.** 30        {(1, 2): 12, (4, 2, 1): 10, (1, 2, 4): 8} **C.** 47     {(1, 2): 12, (4, 2, 1): 10, (1, 2, 4): 8} **D.** 30     {[1, 2]: 12, [4, 2, 1]: 10, [1, 2, 4]: 8} |

**Question 8:** There is a function given and the function is called using some arguments. What is the output after running the entire code? Circle the option that you select. (**10 points : 2 points each**)

|  |  |  |
| --- | --- | --- |
| **No.** | **Code** | **Result** |
| **1** | def ends(str):  if len(str) < 2:  return ' '  return str[0:2] + str[-2:]  print(ends('w3resource')) | 1. w3ce 2. wc 3. w3e 4. wrso |
| **2** | def chars\_mix\_up(a, b):  new\_a = b[:2] + a[2:]  new\_b = a[:2] + b[2:]  return new\_a + ' ' + new\_b  print(chars\_mix\_up('abc', 'xyz')) | 1. xyz abc 2. xyc abz 3. abc xyz 4. axb cyz |
| **3** | L = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]  L = [t for t in L if t]  print(L) | 1. [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')] 2. [(), (), ('',), ('a', 'b'), ('a', 'b', 'c')] 3. [('a', 'b'), ('a', 'b', 'c'), ('d')] 4. [ ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')] |
| **4** | fruit = {}  def addone(index):  if index in fruit:  fruit[index] += 1  else:  fruit[index] = 1    addone('Apple')  addone('Banana')  addone('apple')  print (len(fruit)) | 1. 1 2. 3 3. 2 4. 4 |
| **5** | def new():  arr = {}  arr[1] = 1  arr['1'] = 2  arr[1] += 1  sum = 0  for k in arr:  sum += arr[k]  return (sum)  new() | **1.** 1  **2.** 2 **3.** 3  **4.** 4 |

**Question 9:** What will be the output of the following code. Write your answer in the box provided on the right. (**5 points – 1 point each**)

|  |  |
| --- | --- |
| **Code** | **Solution** |
| S={1,2,3}  S.intersection\_update({5,6,7,8,9})  S | {5,6} |
| {1,2,3,5} | {2,3,4,5,6} | {1,2,3,4,5,6} |
| a={0,1,2,3}  b={0,2,3,4}  a & b | {0,2,6,12} |
| {-1,-2,1,2,3,5} ^ {-1,-2,-3,2,3,4,5,6} | {-3,-2,-1,1,2,3,4,5,6} |
| Y={1,2,3}  Y.add(4)  Y | {1,2,3,4} |

**Question 10**: Write the Output of each program given, in either ‘True’ or ‘False’. (**5 points – 1 point each**).

|  |  |
| --- | --- |
| **Code** | **Solution** |
| x=True  y=True  x and False == False | True |
| x=True  y=False  x or False == x | False |
| not x == x | False |
| init\_tuple\_a = 'a', 'b'  init\_tuple\_b = ('a', 'b')  print (init\_tuple\_a == init\_tuple\_b) | False |
| a = {'a':1,'b':2,'c':3}  b = {'b':2,'a':1,'c':3}  a!=b | True  Doesn’t take into consideration order |