**Instructions:** The following exam contains 25 questions from IIPP, PoC, and AlgTh. We suggest that you browse all of the questions first before you start working seriously on the exam. Also, please review the Honor Code for this exam before you begin. You may use any materials available on the web. But, you may not solicit help from another human to complete this exam.

You have an unlimited numbers of attempts at this exam. However, this is a three day delay between attempts. Each time you submit the exam for grading Coursera will select new variants of the exam questions. Note that the differences between these variants can be small. **Work through each exam question carefully on each attempt.**

Finally, if you are convinced that an exam question is in error, you may send email to *fundamentalscomputing@online.rice.edu*. However emails of the form "I'm confused about a problem" may be ignored. Good luck on the exam. You can do this!

**Question 1:** Consider a function in Python whose execution terminates with the statement



1

return

Enter the value returned by the function in the text box below.



Question 2

1  
point

**2. Question 2**

Consider the following snippet of Python code:



1

2

3

4

5

6

7

8

9

var1 = 7

def var0(var1, var2):

var3 = var1 + var2

global var4

var4 = 17

return var3 + var4

print var0(var1, var1)

What global names are created during execution of this code snippet? What local names are created during execution of this code snippet?



**Global** - \color{red}{\verb|var0|}var0, \color{red}{\verb|var1|}var1, \color{red}{\verb|var4|}var4

**Local** - \color{red}{\verb|var2|}var2, \color{red}{\verb|var3|}var3



**Global** - \color{red}{\verb|var1|}var1, \color{red}{\verb|var4|}var4

**Local** - \color{red}{\verb|var1|}var1, \color{red}{\verb|var2|}var2, \color{red}{\verb|var3|}var3



**Global** - \color{red}{\verb|var0|}var0, \color{red}{\verb|var4|}var4

**Local** - \color{red}{\verb|var1|}var1, \color{red}{\verb|var2|}var2, \color{red}{\verb|var3|}var3



**Global** - \color{red}{\verb|var0|}var0, \color{red}{\verb|var1|}var1, \color{red}{\verb|var4|}var4

**Local** - \color{red}{\verb|var1|}var1, \color{red}{\verb|var2|}var2, \color{red}{\verb|var3|}var3

Question 3

1  
point

**3. Question 3**

Which of the following Python expressions can be used as a key to a dictionary in Python?



\color{red}{\verb|[0]|}[0]



\color{red}{\verb|0|}0



\color{red}{\verb|set([0])|}set([0])



\color{red}{\verb|(0)|}(0)



\color{red}{\verb|False|}False



\color{red}{\verb|"0"|}"0"

Question 4

1  
point

**4. Question 4**

In \color{red}{\verb|SimpleGUI|}SimpleGUI (and most other GUIs), a point on the canvas is indexed by two coordinates. Which statement below correctly characterizes the change in the position of a point on the canvas as these coordinates are varied?



Increasing the first coordinate moves the point downward. Increasing the second coordinate moves the point right.



Increasing the first coordinate moves the point right. Increasing the second coordinate moves the point downward.



Increasing the first coordinate moves the point upward. Increasing the second coordinate moves the point right.



Increasing the first coordinate moves the point right. Increasing the second coordinate moves the point upward.

1  
point

**5. Question 5**

We will revisit the \color{red}{\verb|BankAccount|}BankAccount class from Quiz 6a in IIPP. Here is a slightly modified template for the \color{red}{\verb|BankAccount|}BankAccount class.



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

class BankAccount:

def \_\_init\_\_(self, initial\_balance):

"""

Creates an account with the given balance.

"""

…

def deposit(self, amount):

"""

Deposits the amount into the account.

"""

…

def withdraw(self, amount):

"""

Withdraws the amount from the account.

Each withdrawal resulting in a balance of

less than 10 dollars (before any fees) also

deducts a penalty fee of 5 dollars from the balance.

"""

…

def get\_balance(self):

"""

Returns the current balance in the account.

"""

…

def get\_fees(self):

"""

Returns the total fees ever deducted from the account.

"""

…

The \color{red}{\verb|deposit|}deposit and \color{red}{\verb|withdraw|}withdraw methods each change the account balance. The \color{red}{\verb|withdraw|}withdrawmethod also deducts a fee of 5 dollars from the balance if the withdrawal (before any fees) results in a balance of less than 10 dollars. Since we also have the method \color{red}{\verb|get\_fees|}get\_fees, you will need to have a variable to keep track of the fees paid.

Implement that \color{red}{\verb|BankAccount|}BankAccount class as described above. Here's one possible test with multiple accounts. This test should print the values 10, 5, 0, and 5.



1

2

3

4

5

6

7

account1 = BankAccount(10)

account1.withdraw(15)

account2 = BankAccount(15)

account2.deposit(10)

account1.deposit(20)

account2.withdraw(20)

print account1.get\_balance(), account1.get\_fees(), account2.get\_balance(),

  account2.get\_fees()

Copy-and-paste the following much longer test. What four numbers are printed at the end? Enter the four numbers, separated only by spaces.



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

account1 = BankAccount(20)

account1.deposit(10)

account2 = BankAccount(10)

account2.deposit(10)

account2.withdraw(50)

account1.withdraw(15)

account1.withdraw(10)

account2.deposit(30)

account2.withdraw(15)

account1.deposit(5)

account1.withdraw(20)

account2.withdraw(15)

account2.deposit(25)

account2.withdraw(15)

account1.deposit(10)

account1.withdraw(50)

account2.deposit(25)

account2.deposit(25)

account1.deposit(30)

account2.deposit(10)

account1.withdraw(15)

account2.withdraw(10)

account1.withdraw(10)

account2.deposit(15)

account2.deposit(10)

account2.withdraw(15)

account1.deposit(15)

account1.withdraw(20)

account2.withdraw(10)

account2.deposit(5)

account2.withdraw(10)

account1.deposit(10)

account1.deposit(20)

account2.withdraw(10)

account2.deposit(5)

account1.withdraw(15)

account1.withdraw(20)

account1.deposit(5)

account2.deposit(10)

account2.deposit(15)

account2.deposit(20)

account1.withdraw(15)

account2.deposit(10)

account1.deposit(25)

account1.deposit(15)

account1.deposit(10)

account1.withdraw(10)

account1.deposit(10)

account2.deposit(20)

account2.withdraw(15)

account1.withdraw(20)

account1.deposit(5)

account1.deposit(10)

account2.withdraw(20)

print account1.get\_balance(), account1.get\_fees(), account2.get\_balance(),

  account2.get\_fees()



Question 6

1  
point

**6. Question 6**

In IIPP, we used reference diagrams to visualize the behavior of Python programs that involved mutable objects such as lists. These reference diagrams can be viewed as instances of directed graphs (ala Algorithmic Thinking) in which nodes contain the data stored in the list and the directed edges correspond to references in the diagram.

If we view reference diagrams as directed graphs in this manner, which of the following snippets of Python code has a reference diagram whose corresponding directed graph contains a cycle?





1

2

crazy = [1, 1]

crazy[1] = crazy[0]





1

2

crazy = [1, 1]

crazy = crazy[1]





1

2

crazy = [1, 1]

crazy[1] = crazy[1]





1

2

crazy = [1, 1]

crazy[1] = crazy

1  
point

**7. Question 7**

Consider this [submitted solution](https://www.coursera.org/learn/fundamentals-of-computing-capstone/resources/OaZzd) to the Rock-paper-scissors-lizard-Spock mini-project from IIPP. This solution is either correct or has one line of code that is in error.

If the program is correct, enter the number 0 in the box below. If the submitted program has an error, enter the number of the erroneous line as a positive integer in the box below.

Note that the lines are numbered starting at one as done in IDLE and CodeSkulptor. (Hint: IDLE displays the line number of the currently selected line in the lower right corner of the window.)



1  
point

**8. Question 8**

Consider this [submitted solution](https://www.coursera.org/learn/fundamentals-of-computing-capstone/resources/VaYrc) to the Pong mini-project from IIPP. Note that the paddles do not move in response to key presses.

Modifying four consecutive lines of code fixes this problem and yields a working program. Enter the line number of the first line of code that needs to be modified. Again, the lines are numbered starting at one as done in IDLE or CodeSkulptor.



1  
point

**9. Question 9**

Consider this [submitted solution](https://www.coursera.org/learn/fundamentals-of-computing-capstone/resources/1MQK0) to the Blackjack mini-project from IIPP. Note that the program throws an \color{red}{\verb|AttributeError|}AttributeError.

Modifying exactly one line of the program corrects this error and yields a program that works correctly. Enter the number of the line of code that needs to be modified. Again, the lines are numbered starting at one as done in IDLE or CodeSkulptor.

