Computer Science 217

Midterm Exam

October 28, 2015

First Name: ____________________

Last Name: _____________________

ID: ___________________________

Class Time (Circle One): 1:00pm 3:00pm

Instructions:

- Neatly print your names and ID number in the spaces provided above.
- Pick the **best answer** for each multiple choice question.
- Answer each question by writing the correct answer in the space provided. **Answer all multiple choice questions using UPPER CASE letters.**
- This exam consists of 14 pages, including the cover. Before answering any questions count the pages and ensure that they are all present.
- You have 1 hour 30 minutes to complete this exam.
- Unless noted otherwise, each question is worth one mark.
- This exam is closed book. You are not permitted to use any electronic devices or reference materials.
- **DO NOT TURN PAST THIS PAGE UNTIL YOU ARE INSTRUCTED TO BEGIN**

This test will be returned in class on Friday October 30th with additional questions. You may earn up to one third of a letter grade (for example, a C would become a C+, a B+ would become an A-, etc.) in additional credit on the midterm exam by completing these additional questions and submitting your answers by Friday November 6th.

1. (12 marks) Consider a hypothetical city where the roads are laid out in a perfect grid pattern. All of the avenues run east-west while all of the streets run north-south. Both the avenues and the streets are numbered with consecutive integers. In this particular city, both positive and negative integers are used. The smallest avenue number is at the south edge of the city, with the numbers increasing as you move north. The smallest street number is on the west edge of the city, with the numbers increasing as you move east.

   Write a program that reads pairs of integers from the user. The first integer in each pair will be an avenue number. The second integer in each pair will be a street number. After reading a pair of values your program should provide instructions for walking from the intersection of 0 avenue and 0 street to the intersection entered by the user. Your program should first tell the user how many blocks to walk north or south. Then it should display the number of blocks to walk east or west. Each number of blocks must be expressed as a positive integer.

   After displaying the walking instructions, your program should display the straight line distance between the intersections computed using Pythagorean Theorem. Recall that the Pythagorean Theorem states that the straight line distance, *c*, between two locations is computed as $c = \sqrt{a^2 + b^2}$ where *a* is the north-south distance and *b* is the east-west distance.

   Your program should continue reading pairs of values and displaying the walking instructions until the user enters 0 for both the avenue number and the street number. Do not display any output after this occurs.

   Sample input and output is shown below. Values entered by the user are shown in bold. The messages in your program should match this sample output.

```
Enter the number of the avenue: 1
Enter the number of the street: 1
Walk 1 block(s) North
Walk 1 block(s) East
The straight line distance is 1.4142135623730951 blocks
Enter the number of the avenue: -2
Enter the number of the street: -3
Walk 2 block(s) South
Walk 3 block(s) West
The straight line distance is 3.605551275463989 blocks
Enter the number of the avenue: 0
Enter the number of the street: 0
```

   Your program should use appropriate Python syntax and meaningful variable names. It does **not** need to include any comments. However, you are free to include them if you find doing so helpful. Do **not** define any functions when completing this question.

   Place your answer to this question on the next page. Do **not** answer the question at the bottom of this page.

0

Place your answer to question 1 on this page.

2. Which of the following items was first used to assist with mathematical calculations?
   A. Abacus
   B. Babbage difference engine
   C. Relay
   D. Transistor
   E. Vacuum tube

Answer: _____

3. Which of the following statements about Moore's Law is most correct?
   A. Moore's law held for approximately 10 years before technological changes and physical limitations made it obsolete.
   B. Moore's law stated that the complexity of computers would double every 10 years.
   C. Moore's law predicted that the cost of a computer would double every year, or perhaps every two years.
   D. More than one of the above statements is correct
   E. None of the above statements are correct

Answer: _____

4. Which of the following is **not** normally considered to be a subfield of computer science?
   A. Databases
   B. Electrical Engineering
   C. Information Security
   D. Information Visualization
   E. Networking

Answer: _____

5. What component of a modern computer coordinates communication between the CPU and all of the slow speed components, such as the hard disk and mouse? Note that "motherboard" and "bus" are not acceptable answers. You must indicate a component, not the connection to the component.

6. Which level of competence in Bloom's taxonomy is characterized by the ability to simply recall a fact without understanding its underlying importance?

7. The process of converting a sequence of steps into a form that can be executed by a computer is known as:
   A. Algorithm development
   B. Application
   C. Complexity reduction
   D. Programming
   E. Top down design

Answer: _____

8. A memory location that holds a value that may change as the program runs is known as a(n):
   A. assignment
   B. float
   C. relay
   D. transistor
   E. variable

Answer: _____

9. Which of the following is an advantage of running a program with an interpreter compared to using a compiler?
   A. Using an interpreter typically makes it easier to run your program on several different types of computers or operating systems.
   B. Using an interpreter typically makes it more difficult for a competitor to copy the algorithms used in your program.
   C. Using an interpreter typically provides better performance when the program is running.
   D. More than one of the above answers is correct
   E. None of the above answers is correct

Answer: _____

10. Consider the following code segment:

```
x = input("Enter a value: ")
```

When the program is run and the user enters 33, what is the type of the value stored in x?
   A. boolean
   B. float
   C. int
   D. number
   E. string

Answer: _____

11. Which type of error must be detected by the programmer without the aid of an error message displayed by Python?
    A. A division by zero error
    B. A logic error
    C. A runtime error
    D. A syntax error
    E. A value error

Answer: _____

12. Which of the following statements about Python is most correct?
    A. Adding an integer to an integer will result in an error
    B. Multiplying an integer by a floating point number will result in an error
    C. Subtracting a string from a string will result in an error
    D. Exactly two of the above answers are correct
    E. Answers A, B and C are all correct

Answer: _____

13. By convention, Python programmers use a name that is in ALL CAPS to denote a(n):
    A. Constant value
    B. Error message
    C. Function name
    D. Reserved word
    E. Type name

Answer: _____

14. Consider the following program:

```
x = 3.14159265
print("x is %.4f" % x)
```

Write the output generated by this program in the space below. Note that your output must match what is displayed by Python **exactly**. An answer containing a single extra or missing character will be considered incorrect.

15. What output is generated by the following code segment?

```
a = 7 + 3 ** 3 - 1
print(a)
```

Answer: _____

16. If you were given the first byte (8-bits) of a string, how could you determine if that byte is an ASCII character or the beginning of a multi-byte UTF-8 character? Provide an explanation that is 2 sentences long or less.

17. Convert 2102 base 3 to base 10. Ensure that you place your answer in the space provided.

Answer: _____

18. Convert 99 base 10 to binary. Ensure that you place your answer in the space provided.

Answer: _____

19. (2 marks) Convert 514 base 6 to base 12. Ensure that you place your answer in the space provided.

Answer: ______

20. A quantity of 4 bits is sometimes referred to as a nibble. How many different values can be represented by a nibble?

Answer: ______

21. Complete the truth table for the expression (A and not B) or (A and B). **Draw a box around the column that represents your answer.**

A	B	
0	0	
0	1	
1	0	
1	1	

22. (2 marks) Fizz-Buzz is a famous small computer programming problem. In this particular variant, the program is supposed to print `Fizz-Buzz` if the number entered by the user is evenly divisible by both 3 and 5. For any number that is not evenly divisible by 3 and not evenly divisible by 5 the program should display the number entered by the user. The program is supposed to print `Fizz` (without anything else) if the number is evenly divisible by 3 and not evenly divisible by 5, and it is supposed to print `Buzz` (without anything else) if the number is evenly divisible by 5 and not evenly divisible by 3.

    The following program correctly solves this problem, except for the two conditions that have been omitted. Complete the program by filling in each of the blanks with the necessary Python code. You are **not** allowed to change the provided code.

```
x = int(input("Enter an integer: "))

if _____________________________________________________:
  print("Fizz-Buzz")
elif ___________________________________________________:
  print(x)
elif x % 3 == 0:
  print("Fizz")
elif x % 5 == 0:
  print("Buzz")
```

Consider the following code segment:

```
a = int(input())
b = int(input())

if a > 0 and b < 0:
  a = a - 1
  b = b + 1
  print(a)
elif a > 0 or b < 0:
  b = b + 1
  print(b)
if b > 0 or a < 0:
  a = a – 1
  print(a)

print(b)
```

23. What output is generated if the user enters 1 for a and 1 for b? ______________________________

24. What output is generated if the user enters -2 for a and 3 for b? ______________________________

25. What output is generated if the user enters 0 for a and -1 for b? ______________________________

Consider the following code segment:

```
a = int(input())
b = int(input())

if a > 0:
  if b < 0:
    a = a – 1
  else:
    a = a + 1
    print(a)
  b = b + 1
elif b > 0:
  if a < 0:
    b = b – 1
  else:
    b = b + 1
    print(b)
  a = a + 1

print(a, b)
```

26. What output is generated if the user enters 1 for a and 0 for b? ______________________________

27. What output is generated if the user enters 0 for a and 2 for b? ______________________________

Consider the following code segment:

```
a = int(input())
b = 3

while b > a:
  b = b - 3

print(b)
```

28. What output is generated by the code segment if the user enters 4 for a? ______________________

29. What output is generated by the code segment if the user enters -4 for a? ______________________

Consider the following code segment:

```
for i in range(0, 5):
  print("X")
  for j in range(i, 5):
    print("X")
print("X")
```

30. How many copies of the letter X are displayed when this code segment runs? ___________________

31. Which of the following code segments contain an example of a nested loop?

A.
```
for i in range(10):
   print(i)
for j in range(10):
   print(j)
```

B.
```
for i in range(10):
   print(i)
   for j in range(10):
      print(j)
```

C.
```
for i in range(10):
   print(i)
while i < 20:
   print(i)
   i = i + 1
```

D. Exactly two of the above answers contain an example of a nested loop

E. Answers A, B, and C all contain an example of a nested loop

Answer: ______

Consider the following code segment:

```
a = int(input())
b = 0
c = 2

while b < a:
  print(c)
  if b % 2 == 0:
    for i in range(0, b):
      c = c * 2
  else:
    c = c – 1
  b = b + 1

print(c)
```

32. What output is generated if the user enters 0 for a? ____

33. What output is generated if the user enters 2 for a? ____

34. What output is generated if the user enters 5 for a? ____

35. (3 marks) The program on the following page is supposed to display all of the integers from 1 up to and including some integer entered by the user followed by a list of each number's prime factors. Numbers greater than 1 that only have a single prime factor will be marked as prime. For example, if the user enters 10 then the output of the program should be:

```
Enter the maximum value to display: 10
1 = 1
2 = 2 (prime)
3 = 3 (prime)
4 = 2x2
5 = 5 (prime)
6 = 2x3
7 = 7 (prime)
8 = 2x2x2
9 = 3x3
10 = 2x5
```

However, the program currently contains 3 different bugs on three different lines that are preventing it from completing its intended task. All of the comments are correct, describing what each line of code is supposed to do. Circle each of the 3 bugs (and only the bug) and write **one sentence or less next to the bug describing what you would need to change to fix it**. Note that in this context a bug means an error that causes the program to crash or compute an incorrect result. It does **not** including stylistic issues like poor variable names.

The text of the program that you are to correct appears on the next page.

Hint: We talked about using `end=""` to prevent Python from moving down to the next line after displaying values briefly during one class. In case you have forgotten, I'd like to remind you that the following line of code prints the letter A without moving down to the next line:

```
print("A", end="")
```

You will see very similar uses of `end=""` in the program on the following page. Since these uses of `end=""` follow the form that I have just reminded you of I would **<u>not</u>** recommend marking them as bugs.

```
# Read the maximum value from the user as an integer
n = int(input("Enter the maximum value to display: "))

# Display an error message and end the program if the number
# entered by the user is too small
if n <= 0:
  print("The maximum value must be a non-negative integer.")
else:
  # Handle 1 as a special case because 1 is not a prime number
  print("1 = 1")

  # For each number from 2 up to and including n
  for i in range(2, n + 1):
    # Display the value of i without moving down to the next line
    print("%d = ", i, end="")

    # f will store factor that we are currently checking
    f = 2

    # j will store the number as we remove factors from it.
    # We will refer to j as the factored number.
    j = i

    # while the factor is less than or equal to the factored number
    while f >= j:
      # while the factored number is evenly divisible by the factor
      while j % f == 0:
        # print the factor without moving down to the next line
        print("%d" % f, end="")

        # if the factor is equal to the original number
        if i == f:
          # mark that it is prime without moving to the next line
          print(" (prime)", end="")

        # remove the factor from the factored number by dividing
        # the factored number by the factor
        j = j / f

        # if the factored number is not 1
        if j != 1:
          # display an multiplication symbol without moving down
          print("x", end="")

      # increase factor by 1
      f = 1

    # Move down to the next line now that all of the factors have
    # been displayed
    print()
```

This page has been left blank intentionally.