CSC 110 Midterm 1 Study Guide

Question 1

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def slope(x1, y1, x2, y2):
  m = (y2 - y1) / (x2 - x1)
  1._____ 2._____(m, 2)

def main():
  slope_1 = slope(5, 5, 10, 10)
  3._____(slope_1)

  slope_2 = slope(1, 4, 3, 4)
  4._____(slope_2)

  slope_3 = slope(0, 0, -3, 4)
  5._____(slope_3)

main()
```

Output:

```
1.0
0.0
-1.33
```

ANSWERS:

1:
2:
3:
4:
5:

Question 2

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def calculate_year_born():
  age_string = 1._____("What's your age? ")
  age = 2._____(age_string)
  year_born = 2023 - age
  message = "You were born in " + 3._____(year_born)
  4._____ message

def main():
  result = calculate_year_born()
  5._____(result)

main()
```

Examples of input and output:

- When the string `42` is entered by the user, the following will be shown in the standard output: `You were born in 1981`
- When the string `23` is entered by the user, the following will be shown in the standard output: `You were born in 2000`

ANSWERS:

1:
2:
3:
4:
5:

Question 3

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def is_square(width, height):
  if width == height:
    1._____ True
  else:
    2._____ False

def main():
  width_string = 3._____("Enter your shape width: ")
  height_string = 4._____("Enter your shape height: ")
  width = 5._____(width_string)
  height = 6._____(height_string)

  if is_square(width, height) == True:
    7._____("Your shape is a square.")
  else:
    8._____("Your shape is a rectangle.")

main()
```

ANSWERS:

1:
2:
3:
4:
5:
6:
7:
8:

Question 4

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def range_of_three(x, y, z):
  max = x

  if y > max:
    max = y

  if z > max:
    max = z

  min = x

  if y < min:
    min = y

  if z < min:
    min = z

  1._____ max - min

def main():
  2._____( range_of_three(3, 2, 8) )
  3._____( range_of_three(10, 2, 8) )
  4._____( range_of_three(10, 2, -8) )

main()
```

Output:

```
6
8
18
```

ANSWERS:

1:
2:
3:
4:

Question 5

Write a function to calculate the slope of a line with two sets of coordinates (x1, y1, x2, and y2) using the formula: $m = \frac{y_2 - y_1}{x_2 - x_1}$ The function should return a float rounded at two decimals representing the slope of a line going through points x_1 y_1 and x_2 y_2.

Test cases (your function definition should work with these function calls):

```
assert slope(5, 5, 10, 10) == 1.0
assert slope(1, 4, 3, 4) == 0.0
assert slope(0, 0, -3, 4) == -1.33
```

Question 6

Write a function to calculate a segment length of a line with two sets of coordinates (x1, y1, x2, and y2) using the formula: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ The function should return a float rounded at two decimals representing the line segment from x_1 y_1 to x_2 y_2

Question 7

Write a function to calculate the area of a ellipse using the the formula: $area = 3.1415 \cdot a \cdot b$ rounding result at two decimals. The function should return a string saying `The ellipse area is X` with X being the calculated ellipse area.

Test cases (your function definition should work with these function calls):

```
assert calculate_ellipse_area(5, 10) == "The ellipse area is 157.08"
```

Question 8

Write a function to calculate the range (the highest minus the lowest) of four numbers.

Test cases (your function definition should work with these function calls):

```
assert range(5, 3, 10, -10) == 20
assert range(1, -4, -3, 4) == 8
assert range(0, 0, -3, 4) == 7
```

Question 9

Write a Python function that takes a numeric argument and returns a **string**: `"Fizz"` if `n` is divisible by 3, `"Buzz"` if `n` is divisible by 5, and `"FizzBuzz"` if `n` is divisible by both 3 and 5 – it returns an **empty string** if the number is not divisible by 3 and not divisible by 5.

You can use the modulus operator (`%`)

Test cases (your function definition should work with these function calls):

```python
assert fizz_buzz(10) == "Buzz"
assert fizz_buzz(12) == "Fizz"
assert fizz_buzz(15) == "FizzBuzz"
assert fizz_buzz(4) == ""
```

Question 10

Write a function to remove duplicate characters in a string. The functions should return a string with unique characters only.

Test cases (your function definition should work with these function calls):

```python
assert remove_duplicates("aeiouaeiou") == "aeiou"
assert remove_duplicates("banana ana banana") == "ban "
assert remove_duplicates("") == ""
```

Question 11

Write python code that takes an input from the user, removes the letter characters from it (to keep only digits) and converts the resulting string to an integer. Your code should have a `main()` function and at least one other function.

Test cases:

- When the user enters `"1a2abd"` the program outputs `12`
- When the user enters `"something50more"` the program outputs `50`

Question 12

Write python code that takes a user input that should be an integer (validate input) and prints out all numbers between zero and the number entered by the user. Your code should have a `main()` function and at least one other function.

Test case:

```
Enter a number: banana
Enter a number: apple
Enter a number: 9
0
1
2
3
4
5
6
7
8
9
```

KEY

Question 1

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def slope(x1, y1, x2, y2):
  m = (y2 - y1) / (x2 - x1)
  1._____ 2._____(m, 2)

def main():
  slope_1 = slope(5, 5, 10, 10)
  3._____(slope_1)

  slope_2 = slope(1, 4, 3, 4)
  4._____(slope_2)

  slope_3 = slope(0, 0, -3, 4)
  5._____(slope_3)

main()
```

Output:

```
1.0
0.0
-1.33
```

ANSWERS:

1: return
2: round
3: print
4: print
5: print

Question 2

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def calculate_year_born():
  age_string = 1._____("What's your age? ")
  age = 2._____(age_string)
  year_born = 2023 - age
  message = "You were born in " + 3._____(year_born)
  4._____ message

def main():
  result = calculate_year_born()
  5._____(result)

main()
```

Examples of input and output:

- When the string `42` is entered by the user, the following will be shown in the standard output: `You were born in 1981`
- When the string `23` is entered by the user, the following will be shown in the standard output: `You were born in 2000`

ANSWERS:

1: input
2: int
3: str
4: return
5: print

Question 3

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def is_square(width, height):
  if width == height:
    1._____ True
  else:
    2._____ False

def main():
  width_string = 3._____("Enter your shape width: ")
  height_string = 4._____("Enter your shape height: ")
  width = 5._____(width_string)
  height = 6._____(height_string)

  if is_square(width, height) == True:
    7._____("Your shape is a square.")
  else:
    8._____("Your shape is a rectangle.")

main()
```

ANSWERS:

1: return
2: return
3: input
4: input
5: float
6: float
7: print
8: print

Question 4

Complete each blank in the code below with ONE of the following Python built-in functions or statements (you won't have to use all of these, some will be used more than once): `print`, `return`, `round`, `input`, `int`, `float`, `str`.

```
def range_of_three(x, y, z):
  max = x

  if y > max:
    max = y

  if z > max:
    max = z

  min = x

  if y < min:
    min = y

  if z < min:
    min = z

  1._____ max - min

def main():
  2._____( range_of_three(3, 2, 8) )
  3._____( range_of_three(10, 2, 8) )
  4._____( range_of_three(10, 2, -8) )

main()
```

Output:

```
6
8
18
```

ANSWERS:

1: return
2: print
3: print
4: print

Question 5

```
def slope(x1, y1, x2, y2):
  m = (y2 - y1) / (x2 - x1)
  return round(m, 2)
```

Question 6

```
def sqrt(n):
  return n**0.5

def segment_length(x1, y1, x2, y2):
  length = sqrt((y2 - y1)**2 + (x2 - x1)**2)
  return round(length, 2)
```

Question 7

```
def calculate_ellipse_area(a, b):
  area = round(3.1415 * a * b, 2)
  message = "The ellipse area is " + str(area)
  return message
```

Question 8

```
def range(a, b, c, d):
  max = a
  if b > max:
    max = b
  if c > max:
    max = c
  if d > max:
    max = d


  min = a
  if b < min:
    min = b
  if c < min:
    min = c
  if d < min:
    min = d

  return max - min
```

Question 9

```
def fizz_buzz(n):
  result = ""
  if n % 3 == 0:
    result += "Fizz"
  if n % 5 == 0:
    result += "Buzz"
  return result
```

Question 10

```
def remove_duplicates(string):
  new_string = ""
  index = 0
  while index < len(string):
    char = string[index]
    if char not in new_string:
      new_string += char
    index += 1
  return new_string
```

Question 11

```
def get_int_from_string(string):
  new_string = ""
  index = 0
  while index < len(string):
    char = string[index]
    if char.isnumeric():
      new_string += char
    index += 1
  return int(new_string)

def main():
  user_input = input("Enter your code:\n")
  result = get_int_from_string(user_input)
  print(result)

main()
```

Question 12

```
def print_all_numbers(number):
  current_number = 0
  while current_number <= number:
    print(current_number)
    current_number += 1

def main():
  user_input = input("Enter a number: ")
  while user_input.isnumeric() == False:
    user_input = input("Enter a number: ")
  input_int = int(user_input)
  print_all_numbers(input_int)

main()
```