

Midterm Exam 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 = 2025 - age
    message = "You were born in " + 3._____(year_born)
    4.____ message

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

main()
```

Program behavior:

What's your age?

42

You were born in 1983

ANSWERS:

1:

2:

3:

4:

5:

Question 3

Write a function to calculate factorial of n using a while loop. Test cases (your function definition should work with these function calls):

```
assert factorial(5) == 120
assert factorial(6) == 720
```

Question 4

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 5

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):

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

Question 6

Continue the above function, write Python code that takes a user input that should be an integer (validate input), if it is not, prompt the user repetitively until entering an integer. Call `fizz_buzz` using the number and return a message. Your code should have a `main()` function and at least one other function.

Program behavior:

```
Enter a number: banana
Enter a number: apple
Enter a number: e
Enter a number: 15
The string is FizzBuzz
```

Question 7

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):

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

Question 8

Write a function to count how many letters are there in a string. The functions should return an integer only. Test cases (your function definition should work with these function calls):

```
assert count_letter("abc1234") == 3
assert count_letter("banana") == 6
assert count_letter("012345") == 0
```

Question 9

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.

Program behavior:

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

Question 10

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.

Program behavior:

```
Enter a string: 1a2abd
12
```

Keys

Question 1

1: return

2: round

3: print

4: print

5: print

Question 2

1: input

2: int

3: str

4: return

5: print

Question 3

```
def factorial(num):  
    answer = 1  
    index = 1  
    while index <= num:  
        answer *= index  
        index += 1  
    return answer
```

Question 4

```
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 5

```

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

```

Question 6

```

def validate_input(user_input):
    if not user_input.isnumeric():
        return False
    return True

def main():
    user_input = input("Enter a number: ")
    while validate_input(user_input) == False:
        user_input = input("Enter a number: ")
    answer = fizz_buzz(int(user_input))
    print("The string is: " + answer)

main()

```

Question 7

```

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 8

```

def count_letter(string):
    index = 0

```

```

counter = 0
while index < len(string):
    if not string[index].isnumeric():
        counter += 1
    index += 1
return counter

```

Question 9

```

def get_answer(int_value):
    index = 0
    while index <= int_value:
        print(index)
        index += 1

def main():
    user_input = input("Enter a number: ")
    while not user_input.isnumeric():
        user_input = input("Enter a number: ")
    get_answer(int(user_input))

main()

```

Question 10

```

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()

```