Foundations in Business Programming
Spring 2026 Final Exam Practice

Instructor: Nathaniel Hobbs
foundations@hobbsresearch.com

Student name: _____ **NetID:** _____

- Read these instructions carefully.
- Do *not* turn this page until you are instructed to do so.
- Fill in your personal info as indicated above.
- This exam is worth **100 points**.
- You may assume that all files mentioned in questions are in the current working directory unless stated otherwise.

Point Summary

Section	Points per Question	Total Points
Multiple Choice (8 questions)	2	16
True/False (8 questions)	2	16
Code Tracing (4 questions)	8	32
Open Ended (3 questions)	12	36
Total		100

1 Multiple Choice Questions

Each question in this section is worth **2 points**.

1.1 What is the value of `len([10, 20, 30])`?

1. 2
2. 3
3. 4
4. Error

1.2 Which expression creates a list containing the squares of the numbers in `nums`?

1. `[n**2 for n in nums]`
2. `[n for n**2 in nums]`
3. `for n in nums:  n**2`
4. `list(n**2, nums)`

1.3 What does `with open("data.txt") as f:` help with?

1. It automatically closes the file when the block is done
2. It converts the file into a dictionary
3. It prevents every possible error
4. It imports the file as a Python package

1.4 What happens when code inside a `try` block raises an error that matches the `except` block?

1. Python stops immediately
2. Python runs the matching `except` block
3. Python ignores the entire program
4. Python restarts the `try` block forever

1.5 Which line correctly imports the `math` package/module?

1. `include math`
2. `import math`
3. `open math`
4. `package math`

1.6 What does `continue` do inside a loop?

1. Exits the loop completely
2. Skips the rest of the current iteration and moves to the next one
3. Deletes the current item from the list
4. Returns a value from the function

1.7 What is printed by this code?

```
D = {"a": 2, "b": 5}
print(D["b"])
```

1. a
2. b
3. 2
4. 5

1.8 Which expression tests whether `x` is greater than 10 and less than 20?

1. `x > 10 and x < 20`
2. `x > 10 or x < 20`
3. `x = 10 and 20`
4. `x > 10 then x < 20`

2 True/False Questions

Each question in this section is worth **2 points**.

2.1 **A list comprehension always creates a list. (True/False)**

2.2 **A dictionary comprehension creates key-value pairs. (True/False)**

2.3 **The `with open(...)` pattern is commonly used so files are closed properly. (True/False)**

2.4 **An `except` block can only run if an exception occurs. (True/False)**

2.5 **The `import` statement allows code from another module or package to be used. (True/False)**

2.6 **A `for` loop can loop over the characters in a string. (True/False)**

2.7 **The expression `=` tests equality. (True/False)**

2.8 **`break` and `continue` do exactly the same thing. (True/False)**

3 Code Tracing

Each question in this section is worth **8 points**.

3.1 What is printed by this code?

```
nums = [1, 2, 3, 4]
out = [n * 2 for n in nums if n % 2 == 0]
print(out)
```

3.2 What is printed by this code?

```
words = ["cat", "horse", "dog", "rabbit"]
count = 0
for w in words:
    if len(w) > 3:
        count += 1
print(count)
```

3.3 What is printed by this code?

```
try:
    x = int("12")
    print(x + 3)
except ValueError:
    print("bad")
```

3.4 Assume `scores.txt` contains exactly these three lines:

```
10
20
30
```

What is printed by this code?

```
total = 0
with open("scores.txt") as f:
```

```
    for line in f:
        total += int(line)
print(total)
```

4 Open Ended Questions

Each question in this section is worth **12 points**.

4.1 Write a function called `positiveSquares` that takes a list of numbers `L` and returns a new list containing the squares of only the positive numbers. You may use a loop or a list comprehension.

4.2 Write a function called `countLines` that takes a filename as input and returns the number of lines in that file. Use `with open(...)`.

4.3 Write a function called `safeInt` that takes a string `s`. It should return `int(s)` if the conversion works, and return `None` if the conversion raises a `ValueError`.