This is a list of topics that may be included in Exam 1. This list is not exhaustive and is only intended to help you review.

Instructions:

1. You are allowed to use one index card of notes (3in x 5in)
2. This is a closed book exam. Do not confer with any other person. Do not use any computer. Do not use any calculator.
3. Show your work. Partial credit will be given. Grading will be based on correctness, clarity and neatness.
4. I suggest that you read the whole exam before beginning to work any problem. Budget your time wisely.

Preparing for the exam:

1. *Study examples and activities*. Make sure you understand every examples and in-class activities.
2. *Read Python programs*. Try the programs from the lectures slides. For each program, go through each instruction and find the expected output. Next, run the same program on your computer and compare the output to your expected output. If there is a match, then you understand the program.
3. *Writing Python programs*. Pick some of the programs that we discussed during the lecture. Write a different solution for the problem. Remember, there is no one way to solve a problem.
4. *Study the lab problems*. Make sure you understand the lab problems and your solutions to the problem.
1 Topics

1. Python keywords (make sure you can identify them)

2. Types of data
   a) Integers: 4, 7, -7
   b) Floats: 4.0, 7.787, -10.993
   c) Strings: ‘hello’, ‘The answer is’
   d) Boolean: True, False

3. Variables: identifiers used to name Python data

4. Strings
   a) Strings are immutable. They cannot be modified.
   b) Slicing (extracting) a string: \texttt{s[i:j:stride]}
   c) You can also use negative indexes to retrieve a value from a string.

5. Arithmetic operator precedence (from highest to lowest)
   a) \texttt{**} (exponentiation)
   b) \texttt{*, /, \%} (multiplication, division, modulo)
   c) \texttt{+, -} (addition, subtraction)

6. Comparison operators (\texttt{<, \leq, >, \geq, =, \neq})
   These operators evaluate to either True or False.

7. Decision-making: if; if-else; if-elif-else

8. The \texttt{+} operator: works in different ways depending on the operands involved.
   a) \texttt{int + int = int} (integer addition)
      Example: 5 + 5 is the integer 10
   b) \texttt{string + string = string} (string concatenation)
      Example: ‘hello’+ ‘goodbye’ is the string ‘hellogoodbye’
   c) \texttt{int + string = error}
      Example: 5 + ‘hello’ is an error

9. The \texttt{*} operator: works in different ways depending on the operands involved.
   a) \texttt{int * int = int} (integer multiplication)
      Example: 5 * 5 is the integer 25
   b) \texttt{string * int = string} (string replication)
      Example: ‘hello’* 4 is the string ‘hellogoodbye’
10. Built-in functions (e.g., float(), input(), int(), len(), print(), str())

11. Decision-making Statements (if, if/else, if/elif/else)

12. Repetition: for loops, while loops

2 Sample questions

Below are sample questions to help you prepare for the exam. Make sure you can solve all of these problems by hand. For most of the questions, you can check your answers by typing in the programs and seeing what happens on the computer.

1. Consider the following code.

```python
x = 1
# Line A
print("A")
if x > 0:
    print("B")
elif x < 0:
    print("C")
else:
    print("D")
print("E")
```

   a) What is the output of this program?
   b) What order are the lines executed?
   c) Suppose Line A is changed to `x = 0`. What is the output of the program?

2. What is the output of the following program?

```python
a = "blueberry"
print("1:", a[0])
print("2:", a[2]*4)
print("3:", a[1:5])
print("4:", a[-1])
print("5:", a[2+3])
print("6:", a[2:])
```

3. For each variable in the code below, provide its data type.

```python
a = 

b = 1
c = 1.0
d = True
e = ""
```

If necessary, you can use the built-in function `type()` to help you determine the datatypes of the variables.
4. Write a program for the Einstein Puzzle using integer division

5. Write a program for the Einstein Puzzle using strings

6. Write a Python program that computes the area of a rectangle. Just in case you forgot, the area of a rectangle is its length times its width. Your program will do the following.
   a) Print a message to the user that the computer will calculate the area of a rectangle.
   b) Prompt the user for the length of a rectangle.
   c) Prompt the user for the width of a rectangle.
   d) Compute the area and then print the result to the screen.

**Example**

```plaintext
Enter the rectangle’s length: 3
Enter the rectangle’s width: 2
Rectangle’s area: 6.0
```

7. Write a program that asks for a given change amount, reports the maximum number of dollars followed by the number of quarters, dimes, nickels, and pennies. Hint: You might find it easier to convert the change amount to cents and then proceeding with your calculations.

**Example**

```plaintext
Enter the change amount: 11.42
11 dollars
1 quarters
1 dimes
1 nickels
2 pennies
```

8. Reconsider Question #7. Write a Python program so that if a denomination has a value of 0, it’s not printed.

**Example #1**

```plaintext
Enter the change amount: 4.32
4 dollars
1 quarters
1 nickels
2 pennies
```
Example #2.

```
1. Enter the change amount: 11
2. 11 dollars
```

Example #3.

```
1. Enter the change amount: 9.43
2. 9 dollars
3. 1 quarters
4. 1 dimes
5. 1 nickels
6. 3 pennies
```

9. Go through each instruction in the program below and find the expected output. Next, run the same program on your computer and compare the output to your expected output.

```
phrase = "Python String"
1. print (phrase[3:5])
2. print (phrase[7:])
3. print (phrase[:6])
4. print (phrase[7:-4])
```

10. Go through each instruction in the program below and find the expected output. Next, run the same program on your computer and compare the output to your expected output.

```
var = "PYTHON"
1. print (var.capitalize())
2. var = "FrontLoader"
3. print (var.lower())
4. print (len(var))
5. str = "e"
6. print (var.count(str))
7. var1 = "Aggies Football Team"
8. print (var1.count("e"))
9. var2 = "Eagle Eyes"
10. print (var2.count("E", 0, 5))
11. var = "Python"
12. print (var.islower())
13. var = "python"
14. print (var.islower())
```
course = "110"
print (course.isdigit())

11. Write a program to compute $\sum_{i=1}^{n} i = 1 + 2 + \ldots n$, where $n$ is specified by the user.

12. Write a program that asks the user to enter a list of integers and then prints a histogram to the screen. The integers are entered as a string separated with spaces. For your program, you can only use the material that we have discussed in class. See examples below.

**Example #1.** The user enters the string $1 \_ 3 \_ 20 \_ 5$ (line 1), where the symbol $\_$ represents a space. The output is then a histogram reflecting the values of the positive integers in the string (lines 3–6).

```
1 Enter a string of positive integers separated by spaces: 1 3 20 5
2 *
3 ***
4 **********
5 ******
```

**Example #2.** The user enters the string $20 \_ 18 \_ 3 \_ 9 \_ 45$ (line 1), where the symbol $\_$ represents a space. The output is then a histogram reflecting the values of the positive integers in the string (lines 3–6).

```
1 Enter a string of positive integers separated by spaces: 20 18 3 9 45
2 *********************************************
3 ******************
4 ****************************
5 ***
6 ********
7 ****************************************************
```