1. What acts as an intermediary between a high-level programming language and the ISA? (1)

2. In the classic parable mentioned in section 2.1, what exactly caused the trouble? How do computer programming languages and algorithms avoid such a situation? (2)

3. Define efficiency and scalability. How are they related? (2)

4. Step through this example program and print x and y at each step (2)
   
   \[ \begin{align*}
   x &= 5 \\
   y &= x + 4 \\
   x &= 3 \\
   x &= 5x + 2(3 - y)
   \end{align*} \]

5. Generally, once one line's instruction is completed, the next instruction to be run is the one on the following line. What type of instructions are an exception to this general rule? (1)

6. What role does the backslash character (\) play in Python syntax? Give an example line of code where it will be useful. (2)

7. What is an operation permitted on strings in Python? Give an example of its usage. (2)
8. How many times will the "hello" be printed by each of the following sequence of statements? What is the value of i at the end in each case?

a. (1)
   i = 0
   while(i < 10):
       print("hello")
       i = i + 1

b. (1)
   i = 3
   while(i <= 10):
       print("hello")
       i = i + 1

c. (2)
   i = 0
   while(i < 2):
       j = 0
       while(j < 3):
           print("hello")
           j = j + 1
       i = i + 1

9. Assuming there is no code before these lines, which lines of Python code would produce errors? If it will produce an error, explain.

   a. x = word
   b. x = "word" + 2
   c. x = "word" + "word"
   d. x = "word" / "word"
10. Write a program to swap (exchange) the values of two variables $a$ and $b$. You may create additional variables if needed. Use the browser-based simulator to write and submit. (2)

11. Write a program to print the first $n$ Fibonacci numbers. (Initialize $n$ as 15). Start with 0 and 1 as the first two numbers. The next number is created by adding the previous two numbers. Thus, the series would go like this: 0 1 1 2 3 5 8 .... Use the browser-based simulator to write and submit. (3)

12. Say you wanted to print out the Celsius equivalent for all integer Fahrenheit temperatures from -50 degrees F to 50 degrees F. Write a program to print out this conversion information. The pseudocode for implementing this is given below.

The equation for converting Fahrenheit ($F$) to Celsius ($C$) is: $F - 32 \times \frac{5}{9}$

i. Set $F$'s initial value to -50 (lower bound)
ii. While $F$ is less than or equal to 50 (upper bound)
iii. Convert Fahrenheit to Celsius.
iv. If Fahrenheit and Celsius are equal
v. print "Fahrenheit and Celsius are equal at -40 degrees!"
vi. Else
vii. print the number of degrees in Fahrenheit followed by the number of degrees in Celsius. This should be printed on four lines with "Fahrenheit:" on the first line, the Fahrenheit value on the second line, "Celsius:" on the third line, and your Celsius value on the fourth line.
viii. Increment $F$

Use the browser-based simulator to write and submit. (3)

Sample output: (do not be concerned with rounding or the number of significant figures)

Fahrenheit:
-50
Celsius:
-45.5556
Fahrenheit:
-49
Celsius:
-45
Fahrenheit:
-48
Celsius:
-44.4444
...