IUSE Knowledge Test

1. Demographic Questions

Your student ID number will be erased from the data file after this survey is linked to the other surveys you have taken or will be taking.

* 1. Please enter your UNL Student ID number.

ID

* 2. This class

3. Gender

4. Year in School

IUSE Knowledge Test

2. Computational Thinking Knowledge Test

1. When solving problems, what are the benefits of using selection statements?

I. A selection statement allows one to alter the control flow of an algorithm or a program.

II. A selection statement allows one to repeat some sequence of operations repeatedly, thus saving the number of lines of code.

III. A selection statement allows one to put in place different actions to be performed under different conditions.

IV. A selection statement allows one to represent logical flow of a solution to a problem in an algorithm.

- ◯ I
- ◯ I, II, & III
- ◯ I & III
- ◯ I, III, & IV

2. When solving problems, what are the benefits of using repetition statements?

I. A repetition statement allows one to control a block of code to be executed for a fixed number of times or until a certain condition is met.

II. A repetition statement allows one to repeat some sequence of operations repeatedly, thus saving the number of lines of code.

III. A repetition statement allows one to represent a loop logically.

IV. A repetition statement allows one to divide a set of operations into different groups to handle different situations.

- ◯ I
- ◯ I & II
- ◯ I, II, & III
- ◯ I, II, III, & IV

3. Which of the following is not true about pre-test and post-test loops?

- While loops are pre-test loops.
- For loops are pre-test loops.
- Do-while (or Repeat-until) loops are post-test loops.
- A post-test loop executes the body of the loop at least once.
- A pre-test loop is exactly the same as a post-test loop in terms of functionality and logic.

4. Which of the following is not true about count-controlled and sentinel-controlled loops?

- For loops are the natural implementation choice for count-controlled loops.
- While loops are the natural implementation choice for sentinel-controlled loops.
- If one knows exactly the number of iterations for a loop, then one should use a count-controlled loop.
- If the number of iterations for a loop depends on a certain condition or event becoming true, then one should use a sentinel-controlled loop.
- A loop cannot be both count-controlled and sentinel-controlled.

5. Given the following code snippet, what are the values of persons[0] and persons[1] (Assume zero-indexing: i.e., the first element of an array is indexed with zero)?

```
int persons [10];
persons[0] = 1;
persons[1] = 2;
int temp = 3;
temp = persons[0];
persons[0] = persons[1];
persons[1] = temp;
```

- 1, 2
- 2, 3
- 1, 3
- 2, 1
- 3, 1

6. Which of the following is not a benefit of using functions in computational problem solving?

- A function is a black box that encapsulates a particular sequence of actions that accomplishes a specific task such that we do not necessarily need to know what those actions are in order to use it – it allows for modularity in problem solving.
- A function can be used simply by knowing what it needs as inputs and what it generates as outputs.
- A function is a mathematical function.
- Functions can be used to break the solution to a problem down into subproblems.
- A function can be reused in different solutions.

7. When we decide whether to use a function, which of the following should we consider?

I. Are there places where you are performing the same action more than once?

II. Are there places where you are performing almost the same action on different inputs?

III. How can you break this problem into smaller pieces? Can you break it down even more?

IV. Are there small problems whose outputs will be used as input to a larger problem?

V. Are there loops?

- I, II, & IV
- I, II, & III
- I, III, & IV
- I, II, III, & IV

8. Why are algorithms necessary in computational problem solving?

I. The concept of algorithm can be used to define the notion of decidability – whether an outcome can be achieved by following a set of steps.

II. An algorithm is a blue-print for the actual implementation of a solution, enabling the conversion of a conceptual solution to a program.

III. Expressing solutions in algorithms allow us to solve problems without having to deal with programming details that might be specific to a particular programming language.

IV. Algorithms are needed for programs to compile.

- ◯ I only
- ◯ I, II, & III
- ◯ III & IV
- ◯ I, II, III, & IV

9. When you consider whether to use sequential search or binary search, which of the following questions should be considered?

I. Is the list of items to be searched sorted?

II. If the list of items is not sorted, is it practical to sort the list before you search it?

III. Is the number of items in the list sufficiently small?

- ◯ I
- ◯ II
- ◯ I & II
- ◯ I, II, & III
- ◯ I & III

10. Why is sorting an important paradigm in computational thinking?

- Sorting allows one to arrange a set of values into a particular order, which in turn facilitate more efficient search of the values.
- Sorting allows one to arrange a set of values into ascending order, which in turn facilitate more efficient search of the values.
- Sorting allows one to divide a set of values into infinitely duplets of values, which in turn facilitate more effective search of the values.
- Sorting allows one to simplify a problem that deals with a list of probably randomly-ordered values to a list of descendingly-ordered values.

11. Which of the following is not true?

- In bubble sort, one compares neighboring elements and swaps those which are not in order.
- In selection sort, one searches through the list for the maximum and swaps the maximum element to the end of the list.
- In selection sort, one only needs to perform one swap on each pass through the list.
- In bubble sort, one only needs to perform one swap on each pass through the list.

12. After two passes of bubble sort, what should the following list be?

3, 9, 8, 6, 4, 1, 10

- 3, 8, 6, 4, 1, 9, 10
- 3, 6, 4, 1, 8, 9, 10
- 3, 8, 9, 6, 4, 1, 10
- 3, 1, 8, 6, 4, 9, 10
- 3, 1, 4, 6, 8, 9, 10

13. What is the output of the following function when foo(3) is called, assuming print() is function that prints its parameter?

```
int foo(int n){
  if(n == 0){
    print(“1”);
    return 0;
  }
  else{
    print(“n”);
    foo(n-1);
  }
}
```

- 3211
- 321
- 3210
- 123