

# CS514 Fall 2022 Exam (Take Home)

**90 points**

Due via Canvas: 11:59 PM Mountain Time, Sunday, October 23, 2022

---

## Honor Pledge

“I have not given, received, or used any unauthorized assistance.”

---

(TYPE YOUR NAME HERE AS A PROXY FOR YOUR SIGNATURE)

---

## Instructions

- Copy and paste the Honor Pledge above into your document and type your name as a proxy for your signature. Note that not signing the honor pledge will not be considered as evidence that a student has committed academic misconduct. Students whose religious tenets prohibit taking oaths will not be expected to sign the pledge. More information on the honor pledge can be found here (<https://tilt.colostate.edu/Integrity/Pledge>).
- Include your name on each page of your answers.
- Questions must be answered in numerical order.
- Do not copy the questions in the document that you will submit. I already have the questions.
- Answers should be succinct, coherent English prose. Make only the main points with necessary support.
- Keep your answers as specific as possible. Avoid generalities.
- All answers including code, graphs, and diagrams must be typed. You can create graphs with any tool that you have access to.
- The exam must be turned in as a single document (PDF only). Code, text, and diagrams must be included in the same document. Separate diagram files will not be accepted.
- Do not post comments and questions about the exam to the course discussion board or Teams. Send questions directly to the instructor via email ([ghosh@colostate.edu](mailto:ghosh@colostate.edu)).
- You may use your notes, books, and available articles, but may not consult with other people, except the instructor. You may not use unauthorized internet sources such as Chegg, CourseHero, etc.
- You must cite your sources properly. Any verbatim quotations must be enclosed in quotation marks, with page numbers indicated. You will receive severe point deductions for using material from the text or other sources without proper citation.
- See the CS department student information guide for guidelines on legitimate and illegitimate consultation.
- Late submissions are not allowed.

**Answer all parts of all nine (9) questions.**

---

1. **(5+5 = 10 points)** (a) Explain the role of measurement in determining the best app for the specific functionality of background removal in a photo editing app available at some app store (e.g., Google Play or Apple's App Store). Precisely define the metrics, mention their scale, and how they would be measured.
- (b) Describe the problems with common measurement schemes used for such apps in the app store and other photo editor app rating websites. Precisely define the metrics, mention their scale, and how they would be measured. Do not make up your own measures. Cite the rating websites or stores where these measures are used.
- 

2. **(5+5 = 10 points)** (a) Using measurement, what meaningful statements can we make about an attribute and the entities that possess it?
- (b) Is it meaningful to talk about doubling an application's usability? If not, how do we compare two different applications in terms of their usability?
- 

3. **(10 points)** Suppose that your team has defined as a goal to "Improve the Effectiveness of the Requirements Elicitation Activities" and has organized a training program for the product managers, product owners, and developers to improve their skills. Use the GQM approach to suggest several relevant questions and measures that will enable you to determine if your team has met their goal.
- 

4. **(1+3+4+4+3 = 15 points)** Your company is about to start a new project to develop a new software system. This project will use (for the first time in the company) test-driven development making use of the JUnit framework to develop the project. The company management has asked you to assess the use of test-driven development on this project and make a recommendation on whether or not to use it in other projects.

Explain how you might assess the costs and benefits of using test-driven development, and what considerations you should make in designing the assessment exercise. Make sure that you describe and justify the (a) type of study that you propose (experiment, case study, survey or combination of methods), (b) any hypotheses, (c) study variables, (d) study plan/design, and (e) how you will analyze the results.

---

5. **(3 points)** Suppose that coverage criterion  $C_1$  subsumes coverage criterion  $C_2$ . Further suppose that test set  $T_1$  satisfies  $C_1$  on program  $P$  and test set  $T_2$  satisfies  $C_2$ , also on  $P$ . Suppose  $P$  contains a fault, and  $T_2$  reveals it. However,  $T_1$  does not reveal the fault. Can this happen? Explain.
- 

6. **(5 points)** Here is a faulty program. Answer questions (a) through (f) using the terminology of error, fault, and failure defined in the textbook.

```
public int countOccurrences (int[] x, int y) {
    // Effects: If x==null throw NullPointerException
    // else return the number of times an element in x equals y.
    int count = 0;
    for (int i = x.length-1; i > 0; i--) {
        if (x[i] == y) {
            count++;
        }
    }
    return count;
}
```

- (a) (1 point) Identify the fault and suggest how to fix it.
  - (b) (1 point) Identify a test case that does not execute the fault.
  - (c) (1 point) Identify a test case that executes the fault, but does not result in an error state. If you think that this is not possible, please explain.
  - (d) (1 point) Identify a test case that results in an error, but not a failure. If you think that this is not possible, please explain.
  - (e) (1 point) For a given test input where  $x$  is the array  $\{5, 3, 5\}$  and  $y=5$ , identify the first error state. Be sure to describe the complete state.
- 

7. **(15 points)** Use Input Space Partitioning to systematically derive test inputs for the `HashMap` class as described in the JDK 19 documentation ([https://download.java.net/java/early\\_access/loom/docs/api/java.base/java/util/HashMap.html](https://download.java.net/java/early_access/loom/docs/api/java.base/java/util/HashMap.html)).

The methods of interest for this question are `containsKey`, `containsValue`, `get`, and `put`. The exact signatures and return types are available in the documentation.

Answer parts (a) through (d) below. You must create an ISP table and a BCC table for each method, in the format shown in the lecture and used in a previous assignment. For your convenience, it is also available here: [https://www.cs.colostate.edu/~cs514dl/2022fa/exams/ISP\\_BCC\\_instructions.pdf](https://www.cs.colostate.edu/~cs514dl/2022fa/exams/ISP_BCC_instructions.pdf)

- (a) (2 points) For each method identify all of the input variables. Don't forget the state variable(s).
- (b) (6 points) Identify several characteristics that suggest partitions. For each characteristic, partition the domain using at least two blocks. Designate one block in each partition as the "Base" block.
- (c) (3 points) Define test input values for the blocks, including the base block.
- (d) (4 points) Define a test input set that satisfies base choice coverage (BCC).

Some input variables along with their characteristics and partitions can be used by multiple methods. Show them in the ISP table of each of those methods.

8. (9 points) Answer the following questions on graph coverage criteria based on the following program fragment.

```
0...  // to check whether n is prime or not.
1...  read n;
2...  //check if n is a multiple of 2
3...  if (n%2==0) return false;
4...  //if not, then just check the odds
5...  for(int i=3;i*i<=n;i+=2) {
6...      if(n%i==0)
7...          return false;
8...  }
9...  return true;
```

- (a) (6 points) Draw a control flow graph for this program fragment. Use this graph to do parts (b)–(e) using the textbook website. While points are not separately assigned for parts (b)–(e), you will need the answers to do parts (f)–(h).
- (b) List the test requirements for node coverage. Use the textbook website to generate the requirements.
- (c) List the test requirements for edge coverage. Use the textbook website to generate the requirements.
- (d) List the test requirements for edge-pair coverage. Use the textbook website to generate the requirements.
- (e) List the test requirements for prime path coverage. Use the textbook website to generate the requirements.
- (f) (1 points) Identify a set of test paths that achieve node coverage but not edge coverage on the graph. If this is not possible, explain why. If you do find a set of test paths that cover all the nodes but not all the edges, list the edges that are not covered.
- (g) (1 points) Identify a set of test paths that achieve edge coverage but not prime path coverage on the graph. If this is not possible, explain why. If you do find a set of test paths that cover all the edges but not all the prime paths, list the prime paths not covered.
- (h) (1 points) Identify a set of test paths that achieve prime path coverage but not edge-pair coverage on the graph. If this is not possible, explain why. If you do find a set of test paths that cover all the prime paths but not all the edge-pairs, list the edge-pairs not covered.

For the last 3 parts, the set of test paths do not have to be the ones generated by the textbook website.

9. **(13 points)** Answer the following logic predicate related questions given the predicate  $p = a \wedge (b \vee \neg c)$ . To make it easy for you to understand what you are doing, and for us to grade your work, use the format that we used in our model solution for a similar question in the last assignment.
- (a) (1 points) Using the definition of predicate and clause used in this course, identify the clauses that are used in predicate  $p$ .
  - (b) (3 points) Compute (and simplify) the conditions under which each of the clauses determines predicate  $p$ .
  - (c) (4 points) Write the complete truth table for all clauses. Label your rows starting from 1, with the clauses in the columns in alphabetical order. You should include columns for the conditions under which each clause determines the predicate, and also a column for the predicate itself.
  - (d) (1 point) List all pairs of rows from your table that satisfy general active clause coverage (GACC) with respect to each clause.
  - (e) (1 point) List all pairs of rows from your table that satisfy correlated active clause coverage (CACC) with respect to each clause.
  - (f) (1 point) List all pairs of rows from your table that satisfy restricted active clause coverage (RACC) with respect to each clause.
  - (g) (1 point) List all 4-tuples of rows from your table that satisfy general inactive clause coverage (GICC) with respect to each clause. List any infeasible GICC test requirements.
  - (h) (1 point) List all 4-tuples of rows from your table that satisfy restricted inactive clause coverage (RICC) with respect to each clause. List any infeasible RICC test requirements.