**University of Michigan – Dearborn**

**Department of Computer and Information Science**

**CIS 285: Software Engineering Tools**

**Fall 2024 Midterm Exam**

**Thu, 10/24, 4:00pm – 5:45pm**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID #: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions:**

* Before answering questions, fill in your name and ID number. Before you turn in your exam, sign below testifying that you have neither given nor received aid on this exam. **UNSIGNED EXAMS WILL NOT BE ACCEPTED!**
* Open book and open note
* No AI tools e.g. ChatGPT etc is allowed during the exam
* No communication tool e.g. discord, is allowed
* Attach cheat sheet to this Word document and only submit this Word document
* Duration: 1 hours 45 mins.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Statement on Academic Conduct:**

The University of Michigan - Dearborn values academic honesty and integrity. Each student has a responsibility to understand, accept, and comply with the university's standards of academic conduct as set forth by the Code of Academic Conduct, as well as policies established by the schools and colleges. Cheating, collusion, misconduct, fabrication, and plagiarism are considered serious offenses. Violations will not be tolerated and may result in penalties up to and including expulsion from the University.

### On Honor, I have neither given nor received aid on this examination

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use Git commands and Github.com to realize the following process. The exam .zip file contains a java file PrimeNumberChecker.java.
   1. Create a new local repository
   2. Add PrimeNumberChecker.java to the repository
   3. Create a new gitHub repository e.g Midterm
   4. Push local to the new GitHub repository
   5. Create a local branch ‘YourName’
   6. Under ‘YourName’ branch, make whatever change in the source code (you may just change integer variable number, the hardcoded number)
   7. Compare the difference of the file content and then commit
   8. Push ‘YourName’ branch to GitHub
   9. Approve and merge the pull request on gitHub
   10. Synchronize local Master with gitbub

* List all the commands in the process. Your command must follow the same order as the process. 15 pts
* Copy/Paste your github repository url here (Don’t make any change to the github repository after the exam. It is a cheat if you do so) 10 pts

1. Assume you are a software project manager and that you’ve been asked to computer earned value statistics for a small software project. At the time that you’ve been asked to do the earned value analysis, 16 tasks have been completed. However, the project schedule indicates that 18 tasks should have been completed. The following scheduling data (in person-days) are available: Compute the schedule variance, cost variance SPI, CPI, and CSI.

25 pts

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Task | Planned Effort | Actual Effort | | 1 | 10 | 12 | | 2 | 8 | 9 | | 3 | 12 | 14 | | 4 | 6 | 7 | | 5 | 7 | 8 | | 6 | 15 | 17 | | 7 | 6 | 8 | | 8 | 3 | 4 | | 9 | 10 | 9 | | 10 | 4 | 3 | | 11 | 6 | 5 | | 12 | 12 | 12 | | 13 | 13 | 13 | | 14 | 8 | 9 | | 15 | 6 |  | | 16 | 5 |  | |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. You are given the following project activities, along with their durations and dependencies:

|  |  |  |
| --- | --- | --- |
| **Activity** | **Duration** | **Predecessors** |
| A | 3 |  |
| B | 4 | A |
| C | 2 | A |
| D | 5 | B, C |
| E | 3 | B |
| F | 2 | D, E |

* 1. Draw the Activity-on-Node (AON) graph for the project. 10 pts
  2. List all possible paths and their durations. 7 pts
  3. Identify the critical path 8 pts

1. Create a Katalon Testing project to automate flight search on aa.com. 25 pts
   1. Create a test case
   2. Record and catch objects in aa.com the process to search round trip flight
      1. From DTW
      2. To NYC
      3. Depart 11/05/2024
      4. Return 11/08/2024
      5. Number of passengers 4
      6. Click on Search
   3. Clean the task by removing unnecessary steps
   4. Rename object whose name is long or contain spaces or you believe not appropriate
   5. Define and initialize variables to represent FromCity, ToCity, DepartureDate, ReturnDate and Number of Passengers
   6. Add FlightData.xlsx to Data file (FlightData.xlsx is included in .zip)
   7. Add For Loop in the scripts to read data from the Excel and do the search.
   8. Copy and paste your final scripts below