**University of Michigan – Dearborn**

**Department of Computer and Information Science**

**CIS 285: Software Engineering Tools**

**Fall 2025 Midterm Exam**

**Thursday, 10/23, 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 cell phone is allowed,  You may use cell phone at the end to take picture of answers you wrote on paper and attach them to the submission.
* 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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use Git commands and Github.com to realize the following process. The exam .zip file contains a java file factorialcalculator.java
   1. Create a new local repository
   2. Add factorialcalculator.java to the repository.
   3. Create a new gitHub repository e.g Midterm
   4. Push local to the new GitHub repository (the Github repository must be public)
   5. Create a local branch ‘YourName’
   6. Under ‘YourName’ branch, make whatever change in factorialcalculator.java (you may just change integer num’s value)
   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. Continuing in question 1, Use VSCode, GitHub, and Jenkins to realize the following scenario.
   1. Open VSCode and choose Open Folder and navigate to Q1’s folder
   2. Under local master branch, use VS Code to modify factorialcalculator.java (You may just change integer num’s value)
   3. In VSCode, commit to local master, then push to github
   4. Connect to campus VPN and login to

<http://141.215.80.219:8080>

user name: cis285

password: cis285

* 1. Create a new freestyle project with the name “YourName\_ factorialcalculator”
  2. Config the project to connect your Midterm github repository
  3. Schedule Build Triggers to poll SCM every minute.
  4. Use the following command in Build/Execute Windows batch command

javac factorialcalculator.java

java factorialcalculator

* Repeat step b and c, then have 2 successful builds in Jenkins. Provide these two Console Output screenshots (Don’t make any change to your Jenkins project after the exam. It is a cheat to do so) 25 pts

1. As a software project manager, you have been asked to compute earned value (EV) statistics for a small software project. At the time of this earned value analysis, 14 tasks have been completed, but according to the project schedule, 16 tasks should have been completed.

The following scheduling data (in person-days) are available:  
Compute the Schedule Variance (SV), Cost Variance (CV), Schedule Performance Index (SPI), Cost Performance Index (CPI), and Cost Schedule Index (CSI). (25 pts)

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

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