Queens College, CUNY, Department of Computer Science

Software Engineering

CSCI 370

Fall 2018

Instructor: Dr. Sateesh Mane

September 2, 2018

1 Policies for projects

- Students may work alone (solo) or may form teams to work on projects.
- **The names of all team members must be submitted to me in advance.**
  1. Maximum of five students in a team.
  2. Team members can be in different sections (afternoon and night sections).
  3. **A student may not be a member of more than one team.**
     This includes "working solo" and "being in a team" at the same time.
  4. If two or more students submit duplicate work **and do not inform me in advance** they are in the same team, it will be treated as cheating.
  5. **If two or more teams submit duplicate work, it will be treated as cheating.**
- **I shall assume a team remains in place until the end of the semester.**
  1. If the composition of a team changes, **then I must be notified immediately.**
  2. *If a team dissolves for any reason,* **then I must be notified immediately.**
  3. **A student may not change teams "mid-project" (including going solo).**
  4. The composition of a team may change only **after a project is completed and before a new project is started.**
- **Progress reports & final writeup.**
  1. **Reports should be submitted once a week, due every Sunday at noon.**
  2. Documents (including code) should be submitted in a zip archive (details to be decided).
  3. For progress reports, a team may submit one progress report for the entire team.
  4. **For the final writeup, each member of a team must submit a report.**
- **Every team member must be able to explain every aspect of the project.**
  1. Oral presentations in class about the current status of the project will be required.
  2. Every team member must be able to give an oral presentation in class.
  3. At my discretion, I may ask one team member to speak on one aspect of the project and another team member to speak on a different aspect of the project.
  4. **Every team member must keep up to date with the project development.**

2 Naming convention for reports

- **Please employ the following naming convention for reports.**
- For progress reports, one team member should submit a progress report for the entire team.

```
StudentId_first_last_CS370_Project[1,2,...].zip          (if working alone)
StudentId_first_last_team_CS370_Project[1,2,...].zip     (for a team)
```

- **For a team, the cover document should list the names of the team members. (See Sec. 3.)**
- **For the final writeup, each member of a team must submit a report.**

```
StudentId_first_last_CS370_Project[1,2,...]_final_report.zip
```

- **Do not encrypt the archive.**

3 Format for reports

- The current plan is that a report consists of a zip archive with multiple text and code files.
- **Do not encrypt the archive.**
- Acceptable file formats for text documents are txt/pdf/docx.
- **The report must contain ALL the project documentation, even if an item was not changed from the previous report.** Basically, each report supersedes the previous reports. It is too complicated to deal with "design schematic in report 1" and "code in report 2" and "revisions to code in report 3" etc.
- "Cover"
    1. **A "cover" document (cover.[txt/pdf/docx]) which lists the updates since the previous report.**
    2. The cover document should list the names of the team members.
    3. **It is acceptable if the cover says "no update" since the previous report.**
    4. It is possible to be stuck on a difficult problem which requires more than a week to solve.
- "Mission statement" (mission.[txt/pfd/docx]). A statement of the "high level" goals of the project. In practice this is probably a copy of what I tell you is the outline of the project. It is just to make sure there is no misunderstanding what the project is all about.
- "Architecture"
    1. A design schematic of how you/team intend to implement the project.
    2. The schematic may contain diagrams/flow charts to describe the software architecture.
- "Code"
    1. Status report of code development of each component (client or server or database, etc.).
    2. Debugging tests.
    3. *If bugs were found, how were they found and how were they fixed.*
- "Milestones" and/or "System integration"
    1. A list of milestones the project should achieve.
    2. *If bugs were found (this may happen during system integration), procedures to debug and retest the code.*
    3. **If a bug is sufficiently serious that a project redesign is required, say so.**
        (a) A "bug" is not always a programming error.
        (b) A bug can be a flaw in the overall design.
        (c) An item was overlooked or a scenario was not anticipated, requiring a redesign.
        (d) This happened in the Apollo moon mission, for example. Three astronauts died.
        (e) *Be honest about it.*