MacEwan UNIVERSITY

CMPT 201 – Assignment 3

MacEwan UNIVERSITY

Overview

- Working in a team
- Assignment 3

MacEwan UNIVERSITY

Working in a Team

- **All** the source code of the project and anything that is required to build it is under GitLab.
- **Anyone** should be able to build the entire project.
- Do not add files not needed such as object and binary files to the repository.
- Every member of team should **update** and **commit** to the version control a few times per day ***after*** testing their task enough and ridding it of any trivial bugs.
- **Communication** among team members is crucial. Everyone should be aware of what the others are doing.

MacEwan UNIVERSITY

Team Formation

- Team of 2-3 students
- Groups already created on BB
- Group formation:
  - Find partners and join BB group
  - Students who have not joined a group by the deadline will be randomly added to a group
- Must use vcs
  - Project must be private only the group members should be able to see the project
- Individual contribution will be evaluated

MacEwan UNIVERSITY

Suggested Plan

- Find out who is in your group
  - Use BB
- Establish contact by e-mail
  - Try to set up a first meeting face-to-face or virtual
- On the first meeting
  - Create a private project on vcs
  - Add all group members to the project: Think about roles: all masters?
  - Discuss DB.h
  - Establish a meeting schedule
    - several times a week
  - Establish a general meeting agenda
    - What progress has been made since last meeting?
    - Were all issues from the previous meeting solved?
    - Which are the issues for next meeting?

Suggested Plan

MacEwan UNIVERSITY

- First milestone:
  - Make sure all members understand the requirements and specifications
  - Agree on Work Breakdown
    - Identify roles for group members
    - Assign fair work to each group member
  - finish DB.h
  - agree on timeline
  - schedule regular meetings

Suggested Plan

MacEwan UNIVERSITY

After Milestone 1

- Work in parallel
- Finish Milestone 2
- Create test cases
- Create a structure for the project
  - Design
  - Test
  - Documentation
  - Development
- Create the design document
- Create the testing document
- Create a branch of the project for each member
  - Members will work on their own branches
  - Merge complete work to the master

MacEwan UNIVERSITY

Suggested Plan

- Completed work must be thoroughly tested
  - It helps if the tester is not the developer
  - Bugs need to be communicated and fixed in a timely manner
- Consider integration testing
  - This could be a team testing effort
- Test the final product using the testing document
- Leave at least 3 days for final documentation

MacEwan UNIVERSITY

Assignment 3

- You will create a database for data such as that in PicnicTable.csv, a dashboard for interacting with user, compression and un-compression programs.
- You will implement DB.c and 3 programs: dashboard.c, compressDB.c, and uncompressDB.c. All 4 files will include DB.h.
- Each of dashboard.c, compressDB.c, and uncompressDB.c will have a main function and will call functions in DB.c
- Only global variable allowed is Db, define it in DB.c.
- Your program must work for any table size
- You may assume that values in test files, or entered by user, will be a subset of values given in PicnicTable.csv.

MacEwan UNIVERSITY

DB.c and dashboard

- DB.c
  - contains implementation of all functions in DB.h
  - include <stdarg.h> for variable length argument functions in DB.h
  - Relevant lecture material (check relevant demos as well):
    - dynamic memory allocation
    - pointers
    - file input output
    - string parsing
- dashboard.c:
  - simple menu
  - will call functions in DB.c

DB.c first few lines ..

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>//for variable length args countEntries,
                   //editTableEntry
#include "DB.h"

DataBase *Db;//global variable's (only) definition
```

compressDB.c and uncompressDB.c

- Both take file names as command-line arguments
- compressDB.c: compress data (in .csv file) into output binary file, will call importDB. For each table write, to file,
  - size followed by entries
  - compress int's using bitfields
  - write other variable length fields as is
  - write variable length fields after compressed int's
- uncompressDB.c: uncompresses data in input binary file into a .csv file.
  - must regenerate table in its original form.
  - will call exportDB
- Relevant lecture material (check demos as well)
  - file input output
  - bitwise operations
  - string parsing

MacEwan UNIVERSITY

Words of Advice

- Start early
- Contribute work regularly
- Test frequently
- Communicate constantly