**CS 201 – Accel Intro to CS**

**SPRING 2014**

**FINAL PROJECT GRADING SHEET II**

***DUE BY MIDNIGHT, THURS. 5/1 TO BB***

NAME: \_\_\_\_\_\_ Zhan\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Tonghe \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last First

NAME: \_\_\_\_\_\_Hou\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Xiaocheng\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last First

Your program will be run in Eclipse with the Windows operating system. Please be sure your program will run in that environment. Use the default location for all input/output files.

|  |  |
| --- | --- |
|  | Possible Pts. |
| Description of the group *since Part 1 submission*. Answer the following questions:   1. How often did your group meet?   **Every two days.**   1. Are there assigned roles for each group member? If so, what are those roles?   **There is not a specific role of each of us, we are all thinking about the project, the relationship between classes, and coding.**   1. Describe how much time each group member spent on this part of the project and what each group member contributed.   **Both of us at least spending two and a half days for this project.**  **Tonghe Zhan: processing data, coding.**  **Xiaocheng Hou: do the UML, coding.**  *Please answer c if you are working alone.* | /5 |
| Please give a brief description of what your program does. If this is part of your program documentation you can copy that here.  **The application can read the file which is the crimes information in Chicago. So our program is an application that can access the crime data which contains the case number, the date, the criminal type, and whether the criminal got arrested or not. And we think it is better for the police station to use it. Because they can update the crime data if necessary, if there are new cases happen, they can add it into the dataset, if there are anything wrong with the records, they can delete them. They also can use this application to manage their records, and display it out in the way they want, like search a specific case in the past, or display all the cases that happened in a specific day, or display all the cases that happened in a specific criminal type. What is more, the application also can show the severity of each kind of crime. And if you have a right input, it can calculate the arrest rate for one day, or one case or one criminal type. It is really a powerful crime information management application.** |  |
| Compiles and runs with no run-time errors, menu is *error-proof*  **Yes** | /10 |
| List the input file(s) that are read by your program:  **Please see the attachments. ZhanT\_HouX\_data.csv & ZhanT\_HouX\_menu.txt&ZhanT\_HouX\_type.txt**  Please submit input files zipped with your project; input file should have at least 24 lines of data. **Do not specify directories in your project code;** your program should find the files in the project directories.  Program reads input files correctly and stores data in an encapsulated array(s) | /5 |
| Please attach your *test plan and test plan results* for this program. Use the test plan provided in your final project specifications; if that test plan has been modified please submit the new test plan, 1 pt/box   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Test plan | Test plan data | Test plan results | Actual results | | Menu options: search, sort, add, delete | Search: search a case number, date, crime type, whether arrest  Sort: execute the sort method  Delete: delete one object  Add: add one object | HX153465  “02/16/2014”  Battery  True  Exxcute the selectSort method  Delete the object with the case number: HX153465  Add an object to the end | Show all event includes:  HX153465  “02/16/2014”  Battery  True  Show the sort list by date  The case HX153465 was deleted  The object added to the end | Show all event includes:  HX153465  “02/16/2014”  Battery  True  Show the sort list by date  The case HX153465 was deleted  The object added to the end | | Menu itself: loop, exit, erroneous inputs | Execute the menu options for several times and input -1 to exit | 1. Input 1, battery, 1, robbery and -1 2. Input letter instead of number | 1. Display another menu and the items have “battery”, after input -1,exit 2. Continue to ask user to make choice | 1. Display another menu and the items have “battery” after input -1,exit 2. Continue to ask user to make choice | | Test polymorphism, both to console and to output file | 1. Execute the rate() method 2. Output all data with the rate | 1. Get the rate of line 14 and line 17. 2. All lines | Medium  Serious | Medium  Serious | | /12 |
| Submit a copy of your output file; your output file should demonstrate polymorphism. Your output file should be named appropriately and should be *formatted (columnated), easily readable* *with each test and test results clearly identifiable*. Be sure your output file displays polymorphism with the overridden method that is specific to your application. You can also demonstrate polymorphism with usual class declaration methods.  Name of output file: \_**ZhanT\_HouX\_output.txt**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | /5 |
| What inheritance relationship(s) do you implement in your program? List the attributes of each of the parent and child class(es).  **The inheritance relationship we implement in our program is the crime type and in those crime types which crime contains weapons.**  **Parent class: crimeType**  **Child class: crimeWithWeapons** | /5 |
| Does your program demonstrate polymorphism?  **Yes.**  Overridden method specific to your application: **the String rate method in the two classes**  Overridden methods in usual class declaratios: **toString method, equals method, compareTo method**  Parent class:\_**ZhanT\_HouX\_CrimeType** Child class: **ZhanT\_HouX\_CrimeWithWeapon**  Additional child class(es)? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Identify the class name and the line numbers in your code where we actually see polymorphism (where do you call overridden methods and allow Java to decide at run-time which overridden method to execute?)  **In the CrimeApp, line 125, line 149, line 154, line 166, line 173, line 180, line 187, line194, line 201 you can see the polymorphism of toString method. In the TestApp, you can see the polymorphism of the specific method –rate(),in line 107,line 108** | /5 |
| What association relationship(s) do you implement in your program OTHER than the List class? List the outer class(es) and their attribute(s).  **The class ZhanT\_HouX\_CrimeType and the class ZhanT\_HouX\_CrimeEvent** | /5 |
| Project uses searching and sorting and demonstrates that these are working with appropriate output.. Describe where/how you search and sort in your program.  **In the TestApp we did the sorting and in the CrimeApp we did the searching** | /5 |
| Project implements subclass of List for searches and sorts.  **Yes** | /5 |
| Project adds, deletes as methods in the List class and demonstrates that these are working with appropriate output.  **Yes.** | /5 |
| Project encapsulates data (no da  ta handling in main())  **Yes.** | /3 |
| Project is well coded with good design (neat, easy to read code, follows good programming guidelines with lots of white space, methods for tasks, no large blocks of code, uses object-oriented approach with template classes)  **I think so.** | /10 |
| Classes: All classes complete (constructors, get(), set(), display() methods, is-a, has-a relationships, equals(), compareTo()), Usual List class methods (isThere, isFull, isEmpty, reset(), getNext(), hasNext(),clear(),add(), insert(), delete())  **Of course, we have all of them.** | /5 |
| Documentation: (3 points/ bullet)   * Entire program (explain what program does; anyone reading this should know if this program is the one they need to use) * Each class (explain the role of this class in the project) * Each method (what task does this method perform?) * In-line comments (explains what code does) * All variables (explains what the variable represents) | /15 |
| TOTAL | /100 |

**Please zip all files and name with last names/initials; names should be in all files.** *No demo: -10, did not name classes with team name –2, did not name zip (\*.zip) with team name –2. No email submissions; all submissions must be to BB.*

*Good programming practices are expected; just because the program “runs” or “works” does not guarantee full points.* ***Automatic deduction of 10% for use of java library method (Eg: <ArrayList>, Arrays.sort, etc.);******points will be deducted for not following directions.***

*Any project found to be similar to any other project will automatically be given a 0 for all students involved. Instructor reserves the right to test (verbally or written) any student(s) to demonstrate knowledge of the final project.*

## PHASE 3: PROJECT ANALYSIS (INDIVIDUAL)

Write a brief answer to the following questions.

* Explain your role in the team. List actual tasks assigned and tasks accomplished.
* How many hours did you spend on this project?
* Describe any difficulties you felt the team experienced in working together.
* Describe any difficulties the team experienced in the technical aspects of the project.
* Do you feel you contributed equally to this project? Explain.
* What did you learn from the project?

*Non-submission will be a 10 pt. deduction from the individual’s grade. Indications that an individual did not contribute to the team as expected will result in deduction from that individual’s grade.* ***Each student should contribute to both Part I and Part II submissions; work should be assigned fairly so all team members can contribute equally.***