# *Programming II (420-G20-HR)*

# *Assignment 1 – Inheritance and Writing to Files*

Date assigned: Thursday, February 10, 2022

Test Plan, Class Diagram, Frame Design due: **Wednesday, February 16, 2022 in class**

Completed Assignment due: **Friday, February 25, 2022**

**Learning Objectives**

Upon successful completion of this assignment, the student will be able to:

1. Design test cases from use cases.
2. Design a frame from use cases.
3. Design a class hierarchy from use cases.
4. Create an abstract super class.
5. Create abstract methods.
6. Create a subclass of an abstract super class.
7. Call the super class constructor from a subclass.
8. Override super class abstract methods in a subclass.
9. Write a delimited record to a data file.
10. Use various String methods

**To be handed in:**

1. A hand drawn frame design, test plan for each use case and class diagram should be handed in for review during the class on Wed, Feb. 16. (Use the Assignment 1 test plan template from **Moodle**.)
2. A Java project called ***username\_*G20\_A01\_ContactList** should be zipped and uploaded to **Moodle** andshould include the following in addition to the Java **src** folder:
   1. ***username\_*G20\_A01\_Test\_Plans.docx** containing the corrected test plans for each use case.
   2. A self-assessment for the assignment. (Use the template from **Moodle**.)

**Programming Style:**

Marks will be deducted for poor style. The following criteria will be used:

* All Java source files must be formatted using the Eclipse formatter. (Right-click and select **Source🡪 Format**.)
* Your code must meet all the requirements of the Heritage College Computer Science Code Checklist.

**Organization:**

5 marks of the assignment mark will be for organization. The self assessment must be completed and included in the assignment folder and the assignment must be:

* + handed in to the correct location,
  + be properly named
  + be complete according to the assignment specifications

***Marking Scheme:***

|  |  |
| --- | --- |
|  | **Out of** |
| Test Case Plan for general contact information | 15 |
| Test Case Plan for Add New Business Contact & Personal Contact specific information | 10 |
| Class Diagram – all classes include; correct use of access; inheritance and containment depicted. | 10 |
| Frame Design – all functionality is captured; visually pleasing design | 5 |
| JFrame java code – all frame components work; can add a new personal and business contact; clear functionality; exit functionality; field validation on all fields. | 40 |
| Class hierarchy java code – code divided correctly between classes; no duplication of code between classes; inheritance implemented correctly; default and non-default constructors exist and correctly make use of inheritance; **abstract methods used**. | 35 |
| Contact file code – writes correctly to either personal or business file | 20 |
| Contact number code – stores correctly in contact number file; works correctly when program is exited and then restarted; works across business and personal contacts | 10 |
| Correct execution against requirements; thoroughly tested against test cases. | 15 |
| Organization | 5 |
| **Total** | **165** |

# Problem Specification

Using classes and inheritance, design an online contact list to keep track of the contact information for people that you know living in Canada. Personal contacts include friends and family. Business contacts include people that you know professionally.

Your program must allow the following:

1. Creation of a new contact person, either personal or business, via a frame. The frame should have the ability to create the new contact, to clear the fields in the frame and to quit the program.
2. For a personal contact, the person’s last name, first name, date of birth, address including street, city, province, postal code, e-mail address, cell phone number including area code, home phone number including area code, Twitter handle, Instagram username, and relationship can be entered.
3. For a business contact, the person’s last name, first name, date of birth, business address including street, city, province, postal code, e-mail address, cell phone number including area code, business phone number including area code, company name, department and job title can be entered.
4. When a new contact person is created, the program must assign the person a unique contact identifier. No two people can have the same contact identifier. This number must be stored in a file named **contactIdentifier.txt** when the program is exited so that when the program is restarted, numbering resumes from the next identifier.
5. The format of the contact identifier is x99999, where x is 'P' for a personal contact and 'B' for a business contact. The number (99999) is generated automatically by the program
6. All business contacts must be saved to a delimited file named **businessContacts.txt**. The file format is listed below.
7. All personal contact information must be saved to a delimited file named **personalContacts.txt**. The file format is listed below.

### Your tasks:

1. Create test plans for the **Add New Personal Contact** and **Add New Business Contact** use cases using the template provided on Moodle. The first add new personal contact test case has been done for you. You can separate your test cases into common functionality, personal contact specific tests and business contact specific tests.
2. Design the frame. This can be a hand drawn design.
3. Design the system. Draw a class diagram that includes all classes involved, including your frame class. You **MUST** use **inheritance** and an **abstract class** in this assignment. You **MUST** have appropriate constructors for all classes, and have the subclass call the superclass constructors. Pay attention to the access (private, public, protected).
4. Get approval of your frame design and class design before you start coding. This step is not optional.
5. Code the JFrame interface.
6. Code the rest of the program according to the specifications. Test as you go.
7. Completely test your program using the test plans you developed in the first step.
8. Document your code with comments.
9. As part of the lab on February 24, the class will be testing your program. Your goal is to have tested your program so thoroughly, that no one can break it or find any bugs.

### Data Format:

The format for the business contact file is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Position** | Contents | **Datatype** | **Delimiter** |
| 1 | contactIdentifier | String | ~ |
| 2 | last name | String | ~ |
| 3 | first name | String | ~ |
| 4 | date of birth – day | int | ~ |
| 5 | date of birth – month | int | ~ |
| 6 | date of birth - year | int | ~ |
| 7 | street number and name | String | ~ |
| 8 | city | String | ~ |
| 9 | province | String | ~ |
| 10 | postal code | String | ~ |
| 11 | e-mail address | String | ~ |
| 12 | cell phone number | String | ~ |
| 13 | business phone number | String | ~ |
| 14 | company name | String | ~ |
| 15 | department | String | ~ |
| 16 | job title | String | \n |

The format for the personal contact file is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Position** | Contents | **Datatype** | **Delimiter** |
| 1 | contactIdentifier | String | ~ |
| 2 | last name | String | ~ |
| 3 | first name | String | ~ |
| 4 | date of birth – day | int | ~ |
| 5 | date of birth – month | int | ~ |
| 6 | date of birth - year | int | ~ |
| 7 | street number and name | String | ~ |
| 8 | city | String | ~ |
| 9 | province | String | ~ |
| 10 | postal code | String | ~ |
| 11 | e-mail address | String | ~ |
| 12 | cell phone number | String | ~ |
| 13 | home phone number | String | ~ |
| 14 | Twitter handle | String | ~ |
| 15 | Instagram username | String | ~ |
| 16 | Relationship | int | \n |

The relationship code for the personal contact file is a numeric code representing the following relationships:

|  |  |
| --- | --- |
| **Relationship Code** | **Relationship** |
| **0** | Unknown |
| **1** | Spouse/Partner |
| **2** | Family |
| **3** | Friend |
| **4** | Acquaintance |
| **5** | Neighbour |
| **99** | Other |

A sample personal contact file, is illustrated below:

**P1000~Doe~Jane~06~09~1985~1234 Busy St~Gatineau~Quebec~J8Y6T3~jane.doe@hotmail.com~819-123-4567~n/a~@jdoe~janedoe99~2**

**P1003~Code~Mahatma~31~01~1980~567 Main St~Ottawa~Ontario~K1Y8A8~mahatma@yahoo.com~613-123-4567~n/a~@mcode~mahatma88~4**

A sample business contact file, is illustrated below:

**B1001~Smith~John~07~10~1965~1234 St-Joseph ~Gatineau~Quebec~J8Y6R3~john.smith@cactus.com~819-123-7654~819-987-4757~Cactus~R&D~Software Manager**

**B1002~Moore~Bill~0~0~0~99 Frank** [**Street~Ottawa~Ontario~K2A9A8~billmoore@shopify.com~n/a~613-456-4567~CRA~ITB~Project**](mailto:Street~Ottawa~Ontario~K2A9A8~billmoore@shopify.com~n/a~613-456-4567~CRA~ITB~Project) **Manager**

### Input field Validation

|  |  |  |
| --- | --- | --- |
| **Field** | **Validation** | **Action** |
| Last Name | Must be present | Display pop-up message and allow user to re-enter |
| First Name | Must be present | Display pop-up message and allow user to re-enter |
| DOB – day | May be present | If not present, default to 0. |
| Must be an integer in the valid range for the corresponding month if present. | If present and not numeric in the valid range, display pop-up message and allow user to re-enter. |
| DOB – month | May be present | If not present, default to 0. |
| Must be an integer in the valid range 1 -12 if present | If present and not numeric in the valid range, display pop-up message and allow user to re-enter. |
| DOB – year | May be present | If not present, default to 0. |
| Must be an integer in the valid range 1921-2021 if present | If present and not numeric in the valid range, display pop-up message and allow user to re-enter. |
| Street number and name | May be present | If not present, default to "n/a". |
| City | May be present | If not present, default to "n/a". |
| Province | May be present | If not present, default to "n/a". |
| Postal Code | May be present | If not present, default to “n/a”. |
| Must be in the format of a postal code (letter,number,letter, number,letter,number) if present | If present and not valid format, display pop-up message and allow user to re-enter. |
| e-mail address | Must be present | Display pop-up message and allow user to re-enter |
| Must be in the format of an e-mail address (@ and then a dot after the @) if present | If present and not valid format, display pop-up message and allow user to re-enter. |
| Cell phone number | May be present | If not present, default to "n/a" and ask for confirmation. If not confirmed, allow user to re-enter. |
| Must be in the format of a phone number 111-111-1111 if present | If present and not valid format, display pop-up message and allow user to re-enter. |
| Home phone number | May be present for personal contacts | If not present, default to “n/a”. |
| Must be in the format of a phone number 111-111-1111 if present | If present and not valid format, display pop-up message and allow user to re-enter. |
| Twitter handle | May be present for personal contacts | If not present, default to "n/a". |
| Instagram username | May be present for personal contacts | If not present, default to "n/a". |
| Relationship | Must be present for personal contacts | Display pop-up message and allow user to re-enter |
| Business phone number | May be present for business contacts | If not present, default to “n/a”. |
| Must be in the format of a phone number 111-111-1111 if present | If present and not valid format, display pop-up message and allow user to re-enter. |
| Must be in the format of a phone number 111-111-1111 if present | If present and not valid format, display pop-up message and allow user to re-enter. |
| Company | Must be present for business contacts | Display pop-up message and allow user to re-enter |
| Department | May be present for business contacts | If not present, default to "n/a". |
| Job Title | May be present for business contacts | If not present, default to "n/a". |

## Appendix I –Use Cases

### Add New Personal Contact

New personal contact is added to the personal contact file.

#### Actors/Roles

The student using the contact list

#### Basic Flow

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | 1. The use case starts when the student chooses to create a new personal contact. |
| {Contact Entry} | | | | | 1. The student enters the last name, first name, date of birth, address, e-mail, and cell phone number of the contact. |
|  | | | | | 1. The student chooses the “Personal Contact” option. |
| {Name Validation} | | | | | 1. The system validates the first and last name. |
| {Date of birth Validation} | | | | | 1. The system validates the date of birth. |
| {Address Validation} | | | | | 1. The system validates the address. |
| {E-mail Validation} | | | | | 1. The system validates the e-mail address. |
| {Cellphone validation} |  |  |  |  | 1. The system validates the cell phone number. 2. The system enables the home phone number, Twitter handle, Instagram username and relationship fields. 3. The system disables the ability to add a new Personal Contact or Business Contact and enables the Submit button. 4. The student enters the home phone number, Twitter handle, Instagram username and relationship. 5. The student chooses to submit (save) the contact. |
|  |  |  |  |  |
|  |  |  |  |
| {Personal Contact Entry} | | | |  |
|  |  |  |  |  |
| {Homephone validation} | | | | | 1. The system validates the home phone number. |
| {Twitter validation} | | | | | 1. The system validates the twitter handle. |
| {Instagram validation} | | | | | 1. The system validates the Instagram username. |
| {Relationship validation} | | | | | 1. The system validates the relationship. |
|  | | | | | 1. The system assigns the next available contact number to the new personal contact. |
|  | | | | | 1. The system increments the next available contact number. |
|  | | | | | 1. The system writes the new personal contact to the end of the personal contact file. |
|  | | | | | 1. The system displays a pop-up message containing the contact number that was assigned and a message stating that the personal contact has been added. |
|  | | | | | 1. The system disables the home phone number, Twitter handle, Instagram username and relationship fields. |
|  | | | | | 1. The system disables the Submit button and enables the ability to add a new personal contact or a new business contact. |
|  | | | | | 1. The system clears all the text fields. |
|  | | | | | 1. The use case ends. |

#### Alternative Flows

{Alternate Workflow 1}

1. At {Name Validation}, if the first name or last name is not specified, the system displays an error message stating that the full name must be entered.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 2}

1. At {Date of birth Validation}, if any of the day, month or year fields has not been specified, the system sets the field to 0.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 3}

1. At {Date of birth Validation}, if the month, day or year is not a valid value (month 1 – 12, year 1921-2021, day 1 – 31 depending on the month) then the system displays an error message stating that the date of birth must be numeric and in the appropriate range and clears the erroneous field(s).
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 4}

1. At {Address Validation}, if any of the street, city, province or postal code is not specified, the system sets the street, city, province or postal code to "n/a".
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 5}

1. At {Address Validation}, if the postal code does not follow the postal code format then the system displays an error message stating that the postal code must follow the postal code format and clears the postal code field.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 6}

1. At {E-mail Validation}, if the e-mail address is not specified, the system displays an error message stating that the e-mail address must be entered.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 7}

1. At {E-mail Validation}, if the e-mail address does not contain an @ in the field, and a period at some point after the @, then the system displays an error message stating that the e-mail address is invalid and clears the e-mail address field.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 8}

1. At {Cellphone Validation}, if the cell phone number is not specified, the system sets the cellphone number to "n/a".
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 9}

1. At {Cellphone Validation}, if the cell phone number does not follow the phone number format of xxx-xxx-xxxx then the system displays an error message stating that the cell phone number must follow the phone number format and clears the cell phone number field.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 10}

1. At {Homephone Validation}, if the home phone number is not specified, the system sets the home phone number to "n/a".
2. The use case continues at {Personal Contact Entry} in the Basic Flow.

{Alternate Workflow 11}

1. At {Homephone Validation}, if the home phone number does not follow the phone number format of xxx-xxx-xxxx then the system displays an error message stating that the home phone number must follow the phone number format and clears the home phone number field.
2. The use case continues at {Personal Contact Entry} in the Basic Flow.

{Alternate Workflow 12}

1. At {Twitter Validation}, if the Twitter handle is not specified, the system sets the Twitter handle to "n/a".
2. The use case continues at {Personal Contact Entry} in the Basic Flow.

{Alternate Workflow 13}

1. At {Instagram Validation}, if the Instagram username is not specified, the system sets the Instagram username to "n/a".
2. The use case continues at {Personal Contact Entry} in the Basic Flow.

{Alternate Workflow 14}

1. At {Relationship Validation}, if the relationship is not specified, the system displays an error message stating that the relationship must be entered.
2. The use case continues at {Personal Contact Entry} in the Basic Flow.

{Alternate Workflow 15}

1. At {Relationship Validation}, if the relationship is not a valid relationship (see table), then the system displays an error message stating that the relationship is invalid and clears the relationship field.
2. The use case continues at {Personal Contact Entry} in the Basic Flow.

{Alternate Workflow 16}

1. At any time, if the student clicks clear, the system clears all input fields.
2. The use case continues at {Contact Entry} in the Basic Flow.

#### Post Conditions

1. The personal contact is added to the personal contact file.

### Add New Business Contact

New business contact is added to the business contact file.

#### Actors/Roles

The student using the contact list

#### Basic Flow

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | 1. The use case starts when the student chooses to create a new business contact. |
| {Contact Entry} | | | | | 1. The student enters the last name, first name, date of birth, address, e-mail, and cell phone number of the contact. |
|  | | | | | 1. The student chooses the “Business Contact” option. |
| {Name Validation} | | | | | 1. The system validates the first and last name. |
| {Date of birth Validation} | | | | | 1. The system validates the date of birth. |
| {Address Validation} | | | | | 1. The system validates the address. |
| {E-mail Validation} | | | | | 1. The system validates the e-mail address. |
| {Cellphone validation} |  |  |  |  | 1. The system validates the cell phone number. 2. The system enables the work phone number, company name, department and job title fields. 3. The system disables the ability to add a new Personal Contact or Business Contact and enables the Submit button. 4. The student enters the work phone number, company name, department and job title. 5. The student chooses to submit (save) the contact. |
|  |  |  |  |  |
|  |  |  |  |
| {Business Contact Entry} | | | |  |
|  |  |  |  |  |
| {Businessphone validation} | | | | | 1. The system validates the business phone number. |
| {Company validation}  {Department validation} | | | | | 1. The system validates the company name. 2. The system validates the department name. |
| {Job Title validation} | | | | | 1. The system validates the job title. |
|  | | | | | 1. The system assigns the next available contact number to the new business contact. |
|  | | | | | 1. The system increments the next available contact number. |
|  | | | | | 1. The system writes the new business contact to the end of the business contact file. |
|  | | | | | 1. The system displays a pop-up message containing the contact number that was assigned and a message stating that the business contact has been added. |
|  | | | | | 1. The system disables the work phone number, company name, department and job title fields. |
|  | | | | | 1. The system disables the Submit button and enables the ability to add a new personal contact or a new business contact. |
|  | | | | | 1. The system clears all the text fields. |
|  | | | | | 1. The use case ends. |

#### Alternative Flows

{Alternate Workflow 1}

1. At {Name Validation}, if the first name or last name is not specified, the system displays an error message stating that the full name must be entered.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 2}

1. At {Date of birth Validation}, if any of the day, month or year fields has not been specified, the system sets the field to 0.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 3}

1. At {Date of birth Validation}, if the month, day or year is not a valid value (month 1 – 12, year 1921-2021, day 1 – 31 depending on the month) then the system displays an error message stating that the date of birth must be numeric and in the appropriate range and clears the erroneous field(s).
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 4}

1. At {Address Validation}, if any of the street, city, province or postal code is not specified, the system sets the street, city, province or postal code to "n/a".
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 5}

1. At {Address Validation}, if the postal code does not follow the postal code format then the system displays an error message stating that the postal code must follow the postal code format and clears the postal code field.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 6}

1. At {E-mail Validation}, if the e-mail address is not specified, the system displays an error message stating that the e-mail address must be entered.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 7}

1. At {E-mail Validation}, if the e-mail address does not contain an @ in the field, and a period at some point after the @, then the system displays an error message stating that the e-mail address is invalid and clears the e-mail address field.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 8}

1. At {Cellphone Validation}, if the cell phone number is not specified, the system sets the cellphone number to "n/a".
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 9}

1. At {Cellphone Validation}, if the cell phone number does not follow the phone number format of xxx-xxx-xxxx then the system displays an error message stating that the cell phone number must follow the phone number format and clears the cell phone number field.
2. The use case continues at {Contact Entry} in the Basic Flow.

{Alternate Workflow 10}

1. At {Businessphone Validation}, if the business phone number is not specified, the system sets the business phone number to "n/a".
2. The use case continues at {Business Contact Entry} in the Basic Flow.

{Alternate Workflow 11}

1. At {Businessphone Validation}, if the business phone number does not follow the phone number format of xxx-xxx-xxxx then the system displays an error message stating that the business phone number must follow the phone number format and clears the business phone number field.
2. The use case continues at {Business Contact Entry} in the Basic Flow.

{Alternate Workflow 14}

1. At {Company Validation}, if the company name is not specified, the system displays an error message stating that the company name must be entered.
2. The use case continues at {Business Contact Entry} in the Basic Flow.

{Alternate Workflow 15}

1. At {Department Validation}, if the department name is not specified, the system sets the department to "n/a".
2. The use case continues at {Business Contact Entry} in the Basic Flow.

{Alternate Workflow 16}

1. At {Job Title Validation}, if the job title name is not specified, the system sets the job title to "n/a".
2. The use case continues at {Business Contact Entry} in the Basic Flow.

{Alternate Workflow 17}

1. At any time, if the student clicks clear, the system clears all input fields.
2. The use case continues at {Contact Entry} in the Basic Flow.

#### Post Conditions

1. The business contact is added to the business contact file.