Assignment 1 – A Collection Class

Objectives: *This assignment gives you some experience with designing and writing C++ classes using "big five" (see the textbook), operator overloading, and templates. Also you will learn how to use the command line interface (CLI) and makefiles.*

- (5 points) Create a text file, called README, in which you provide:
  - Your First Name, Last Name, UIN, Section Number, User Name, E-mail address
  - State the Aggie Honor statement:

    I certify that I have listed all the sources that I used to develop the solutions and code to the submitted work.

    > *On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*

    Your Name Date

  - List any resources used such as webpages (provide URL). Do not mention the textbook and discussions with the Instructor, TA, or Peer Teachers.
  - List any known problems with the assignment you are turning in. For example, if you know your code does not run correctly, state that. This should be a short explanation.
  - Provide a short description or pseudocode.
  - Write how you tested your program(s) for correctness and how you used exceptions for the error handling.
  - Submit to eCampus an electronic version of the file README along with your C++ code for the problems listed below and a hard copy to your TA.
  - The assignment will be graded focusing on: program design, correctness.
  - Your programs will be tested on a CSE Linux machine.
  - Provide all tests done to verify correctness of your program in the report. Give an explanation why have you selected such tests.
  - Your program will be tested on a TA's input file.

Problem Description – Part 3 (95 pts)

1. (50 points) Write a templated version of the class `Collection` with the template parameters: `Obj`, `F1`, `F2`.

   (a) The templated class `Collection` and all the templated functions should be in the header file `collection.h` (the file `collection.cpp` is not necessary and can be dropped).

   (b) Test it with the class `Stress_ball` as `Obj`, `Stress_ball_colors` as `F1`, and `Stress_ball_sizes` as `F2`. Use the file `stress_ball_test.cpp`.

   - Perform the same operations: `make_copy`, `make_union`, `swap`, and `sort_by_size` for testing.
   - Modify the test file from Part 2 to do this. In order not to use long class names, use aliases:

```
using CollectionSB = Collection<Stress_ball, Stress_ball_colors,
                                    Stress_ball_sizes>;
using CollectionJN = Collection<Jeans, Jeans_colors, Jeans_sizes>;
```

   (c) The input `operator>>` can be templated but you need to use a specific version for each template class. So for the class `Stress_ball` use this approach:
   `istream& operator>>(istream& is, CollectionSB& c);`
   where you explicitly use the class `Stress_ball` (do not use template parameters `Obj`, `F1`, or `F2`). And do not put it in the file `collection.h` but put in the file `stress_ball_test.cpp`.

2. (30 points) Write a class `Jeans` similar to `Stress_ball` with the same class members: color and size. Feel free to use your own colors and sizes. Suggested colors: white, blue, brown, black, and suggested sizes: small, medium, large, xlarge.

   (a) Apply the `Collection` functions to `Jeans` objects

   (b) Provide similar testing cases for the templated `Collection` with `Jeans` objects.

   - Use the test file `jeans_test.cpp`.

   (c) For the input `operator>>`, use the class `Jeans` explicitly (do not use template parameters `Obj`, `F1`, or `F2`):
   `istream& operator>>(istream& is, CollectionJN& c);`
   Do not put it in the file `collection.h` but use it in the file `jeans_test.cpp`.

3. Here is an example of the file `collection_test.cpp` but you can modify it.

```
#include <iostream>
using namespace std;
void test_stress_balls();
void test_jeans();
int main() {
  int answer;
  cout << "What version to test: stress_ball (=0) or jeans (=1): ";
  cin >> answer;
  if (answer == 0)
    test_stress_balls();
  else if (answer == 1)
    test_jeans();
  else
    cout << "Wrong value: " << answer << endl;
  return 0;
}
```

4. (15 points) Write about the generic programming using templates based on this assignment Part 3.

The templated class Collection and the class Jeans should be presented to your TA or PT during the labs by February 16 and submitted to eCampus. You should test all the implemented functions/operators.