

CS214: Data Structures

Assignment-1

Instructions

- 1- Students will form teams of 2 students (from any group since lab groups are not assigned yet).
- 2- Deadline of submission is **Friday Mar.11th at 11:55 pm.**
- 3- Submission will be on Blackboard.
- 4- No late submission is allowed.
- 5- No submission through e-mails.
- 6- You will write a cpp file with the name firstStudentID_SecondStudentID.cpp .
No rar or zip files are allowed. No folders, just submit a single file.
For example if you have an ID 20202020 and your partner has the ID 19191919, then your cpp file will have the name 20202020_19191919.cpp .
The character separating the ids is an Underscore character. Do not use any other characters like a space, a dash, or a bracket,...etc.
- 7- **In case of Cheating you will get a negative grade whether you give the code to someone, take the code from someone/internet, or even send it to someone for any reason.**
- 8- You have to write clean code and follow a good coding style including choosing meaningful variable names.

Task

Create a class **IntegerSet** for which each object can hold integers in the range 0 through 100. A set is represented internally as an array of ones and zeros. Array element $a[i]$ is 1 if integer i is in the set. Array element $a[j]$ is 0 if integer j is not in the set.

- The default constructor initializes a set to the so-called “empty set,” i.e., a set whose array representation contains all zeros.
- Provide member functions for the common set operations. For example, provide a **unionOfSets** member function that creates a third set that is the set-theoretic union of two existing sets (i.e., an element of the third set’s array is set to 1 if that element is 1 in either or both of the existing sets, and an element of the third set’s array is set to 0 if that element is 0 in each of the existing sets).

CS214: Data Structures

Assignment-1

- Provide an **intersectionOfSets** member function which creates a third set which is the set theoretic intersection of two existing sets (i.e., an element of the third set's array is set to 0 if that element is 0 in either or both of the existing sets, and an element of the third set's array is set to 1 if that element is 1 in each of the existing sets).
- Provide an **insertElement** member function that inserts a new integer k into a set (by setting a[k] to 1).
- Provide a **deleteElement** member function that deletes integer m (by setting a[m] to 0).
- Provide a **printSet** member function that prints a set as a list of numbers separated by spaces. Print only those elements that are present in the set (i.e., their position in the array has a value of 1). Print --- for an empty set.
- Provide an **isEqualTo** member function that determines whether two sets are equal.
- Provide an additional constructor that receives an array of integers and the size of that array and uses the array to initialize a set object.
- Provide sets difference using the **operator –** . For example, **s1 = s2 – s3** will result in s1 having the elements in s2 that are not in s3.
- Provide set complement using the **operator !** . For example, **s2 = !s1** will result in s2 having all of the elements that are not in s1.
- Now write a driver program (main) to test your IntegerSet class.

Instantiate several IntegerSet objects. Test that all your member functions work properly.