

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444**Submission date:****May 31, 2021**

Workshop 1

Description:

The first Assignment lets you practice basic java coding techniques, creating classes, methods and using arrays.

Task 1:

Create class *IntegerSet*, with the followings,

- Each *IntegerSet* object can hold integers in the range **0–100**.
- The set is represented by an array of booleans.
- Array element $a[i]$ is true if integer i is in the set.
- Array element $a[j]$ is false if integer j is not in the set.
- The no-argument constructor initializes the array to the “empty set” (i.e., all false values).

Provide the following methods:

- The static method *union* creates a set that is the set-theoretic union of two existing sets (i.e., an element of the new set’s array is set to true if that element is true in either or both of the existing sets—otherwise, the new set’s element is set to false).
- The static method *intersection* creates a set which is the set-theoretic intersection of two existing sets (i.e., an element of the new set’s array is set to false if that element is false in either or both of the existing sets—otherwise, the new set’s element is set to true).
- Method *insertElement* inserts a new integer k into a set (by setting $a[k]$ to true).
- Method *deleteElement* deletes integer m (by setting $a[m]$ to false).
- Method *toString* returns a String containing a set as a list of numbers separated by spaces. Include only those elements that are present in the set. Use --- to represent an empty set.
- Method *isEqualTo* determines whether two sets are equal.

Write a program to test class *IntegerSet*. Instantiate several *IntegerSet* objects. Test that all your methods work properly including the constructor. One of the possible output for the program

```
Input Set A
Enter number (-1 to end): 10
Enter number (-1 to end): 20
Enter number (-1 to end): 30
Enter number (-1 to end): 50
Enter number (-1 to end): -1
Input Set B
Enter number (-1 to end): 40
Enter number (-1 to end): 60
Enter number (-1 to end): 102
Enter number (-1 to end): 20
Enter number (-1 to end): -1
Set A contains elements:
{10 20 30 50 }
Set B contains elements:
{20 40 60 }
Union of Set A and Set B contains elements:
{10 20 30 40 50 60 }
Intersection of Set A and Set B contains elements:
{20 }
Set A is not equal to set B
Inserting 77 into set A...
Set A now contains elements:
{10 20 30 50 77 }
Deleting 77 from set A...
Set A now contains elements:
{10 20 30 50 }
New Set contains elements:
{1 2 9 25 45 67 99 100 }
```

Students can change the output look as required to make it better in more understandable way.

Task 2:

This task is required to create Craps, which is a popular dice game played in casinos. You are supposed to write a program to play a variation of the game, as follows:

- Roll two dice. (Each roll should produce two random numbers between 1 to 6)
- Each die has six faces representing values 1, 2, ..., and 6, respectively.
- Check the sum of the two dice. If the sum is 2, 3, or 12 (your program should display *craps*), you lose the game.
- If the sum of the two dice is 7 or 11 (your program should display *naturals*), you win the game.
- If the sum of two dice is any value (i.e., 4, 5, 6, 8, 9, or 10), your program should establish a point in the game (meaning store that sum). Continue to roll the dice until the sum is either a 7 or the same point value which was established. If rolled sum is 7, you lose the game. Otherwise, if the rolled sum is equal to established point you win.

Your program acts as a single player.

You rolled $5 + 6 = 11$
Congratulations, You win

You rolled $1 + 2 = 3$
Craps, Better Luck Next Time, You lose

You rolled $4 + 4 = 8$
Point is (established) set to 8
You rolled $5 + 1 = 6$
You rolled $1 + 1 = 2$
You rolled $6 + 2 = 8$
Congratulations, You Win

You rolled $5 + 1 = 6$
Point is (established) set to 6
You rolled $2 + 5 = 7$
Craps, Better Luck Next Time, You Lose

Continue to the next page...

Workshop Header

/*****

Workshop #

Course:<subject type> - Semester

Last Name:<student last name>

First Name:<student first name>

ID:<student ID>

Section:<section name>

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature

Date:<submission date>

*****/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention.
- Document all the classes properly
- Do Not have any debug/ useless code and/ or files in the assignment

Deliverables and Important Notes:

All these deliverables are supposed to be uploaded on the blackboard once done.

- You are supposed to submit a video/ detailed document of your running solution. **(40%)**
 - Things to be considered if you are uploading a video.
 - A reasonable length of video should be posted. 5%
 - Your video should show the running solution with different inputs. 5%
 - In a minute discuss the design of your solution. 10%
 - In a minute (max 3 to 5) discuss the important functions/ methods in your solution. 20%
 - If you are using in your solution concepts that are not discussed in the class then in a minute or two explain,
 - What is that concept?
 - Why did you use it?

- How does it benefit your solution?
- Things to be considered if you are uploaded the detailed document.
 - Should include **screen shots** of your output. 5%
 - Underneath each screen shot explain in 2 to 4 lines what is happening. 10%
 - In 3 to 5 lines explain the design logic of your program. 10%
 - Screen shots of important methods/ functions in your solution and discuss them underneath each screen shot (3 to 6 lines). 15%
 - If you are using in your solution concepts that are not discussed in the class then in a minute or two explain,
 - What is that concept?
 - Why did you use it?
 - How does it benefit your solution?
- A word/ text file which will reflect on learning of your concepts in this workshop. (Also include the instructions on how to run your code, if required) **(30%)**
 - Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali_123456.txt
- Submission of working code. **(30%)**
 - Make sure you follow the “**Code Submission Criteria**” mentioned above.
 - You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.zip**.
- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.
- Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.