

COMP 249: Object Oriented Programming II

Tutorial 8: Binary File I/O

Question 1 - Part 1

❖ Write a Java program that:

1. Reads the provided text file named "chars.txt"
2. Prints the ASCII value of each character in the file.
3. Keeps track of some additional information for each character, i.e., whether it is a letter , a digit, or a special character.

Question 1 - Part 2

❖ Write a Java program that:

1. Reads the provided text file named "chars.txt"
2. Prints the ASCII value of each character in the file
3. Keeps track of some additional information for each character, i.e., whether it is a letter , a digit, or a special character
4. Provide a statistical overview (rounded and in %) of the character categories found in the file and write them to a separate file named "insightsOfChars.txt"

Question 2 - Part 1

❖ Write a Java program that:

1. Prompts the user to enter **ten** integers.
2. Adds them, one integer per line, to a binary file called "inputQuestion2.dat"

Question 2-Part 2

❖ Write a Java program that:

1. Prompts the user to enter **ten** integers.
2. Adds them, one integer per line, to a binary file called "inputQuestion2.dat"
3. Calculates the average of the even integers and the sum of the odd integers in the file.
4. Writes both the average and the sum to a binary file named "outputQuestion2.dat"

Question 3

❖ Write a Java program that:

1. Searches the binary file “*inputQuestion2.dat*” you created in the question 2 and outputs the largest and smallest numbers to the screen.
2. Your program should make sure that there is at least one integer in the file.
3. If the file is empty, it should display a message to that effect and terminate the program.

Question 4 - Part 1

1. Create a simplified class named **SignUp**, with *username* and *password* as the attributes.
2. Implement all the basic needed functions, including constructors, accessors, mutators, etc.
3. Include a method isUsernameValid(String username) to validate the username:
 - a. is between 6 and 10 characters.
 - b. contains only letters and numbers.
 - c. provides an error message otherwise.
4. Include a method isPasswordValid(String password) to validate the password:
 - a. is between 12 and 20 characters.
 - b. contains at least one special character (e.g., @, #, etc.).
 - c. provides an error message otherwise.
5. Create a passwordEncryption(String password) method that uses a simple caesar cipher encryption for letters (shifts each password characters by 3 positions) and keeps digits/special characters unchanged (**MAKE SURE** to wrap around if it goes beyond 'z'/'Z')

Question 4 - Part 2

In a driver class:

1. Create an object of the SignUp class.
2. Allow user to input their credentials and validate them (for a max of **three attempts** per attribute).
3. Using binary IO, write the encrypted SignUp object into a file named "userCredentials.dat."
4. Read and Display (for your own verification) the encrypted credentials from the file.
5. Display the decrypted credentials to the user (□ *another method in SignUp?*). Careful to prompt the user for their password ONCE before presenting their decrypted credentials for security purposes.