Assignment DAY2:

Collections:

1. Use a HashSet to hold Employee Objects. Upon running the application, the details of the employees added to the HashSet should be displayed.

Employee <<class>>

|-- id

|-- name

|-- salary

|-- department

|-- displayDetails()

Feel free to add properties and methods to Employee Class

Note: if we try to store any object other than Employee Object in HashSet, we should not be

allowed to.

Use the same Employee object to add 2 instances of Employee object in a TreeSet.

1. Write a program to validate a phone number using Regex in Java
2. Write an application to perform withdraw functionality on a SavingAccount object. Point to

note:

a. Raise InsufficientBalanceException if you are trying to withdraw more than balance or when you balance is zero. E.g. if you balance is 2000 and if you are trying to withdraw 2100 or if you balance is 0 and you are trying to withdraw positive value.

b. Raise IllegalBankTransactionException if you are trying to withdraw a negative value from your balance. E.g. if you try to withdraw a negative value savingAcc.withdraw(-1000);

Note: SavingAccount

|-- long id

|-- double balance

|--double withdraw(double amount)

|--double deposit(double amount)

1. Problem: Filter Strings by length given a list of Strings, write a Java Program to filter out all strings whose length is less than or equal to given length and using streams.Sample Input: ["apple", "banana", "orange", "grape", "kiwi"], n = 5 Sample Output: ["banana", "orange"]
2. Calculate Factorials. Given a list of integers , write a program to calculate factorial of each number using streams.