Assignments

Collections, Exception Handling, Lambda Expression and Threads

1. Create the application Shopping cart with the given information:

**package** com.demo.assignment2;

**public** **abstract** **class** Item {

**private** **int** itemId;

**private** String itemName;

**private** **double** itemPrice;

//generate required methods

**public** String getItemDetails() {

**return** itemId+"-"+itemName+" : "+itemPrice;

}

}

**package** com.demo.assignment2;

**import** java.util.HashMap;

**public** **interface** ShoppingCart {

**public** HashMap<Integer,Item> displayCart();

**public** String addItem(Item item);

**public** String removeItem();

}

The above class Item has subclasses as:

1. Mobile
2. Shirt
3. Toy

Implement the ShoppingCart interface with a classShoppingCartImpl.

Add the given items to the shopping cart and use various methods to demonstrate the functionality.

1. Create a Mail Editor Application with the given inputs as below:

**package** com.demo.assignment1;

**class** Mail{

**private** String toAddress;

**private** String fromAddress;

**private** Message message;

//generate no-args constructor

//generate all args constructor

//generate setter and getter methods

}

**package** com.demo.assignment1;

**class** Message{

**private** String header;

**private** String body;

//generate the required methods

}

**package** com.demo.assignment1;

**class** MailEditor {

**public** Mail compose(String from, String to, String message, String header) {

// create a Message Object

// Create the Mail Object

// return the Mail Object;

}

**public** **void** send() {

//print out the mail details on console

}

}

1. Consider the following String array

String[] fruits={“apple”,”banana”,”pear”,”mango”,”guava”};

Use streams api to :

1. reverse, sort and capitalize all the fruits from the array in a stream pipeline and display the result.
2. Select all the fruits which contains the letter ‘e’ and display on the console.
3. Consider the following given class:

**class** Owner{  
 **private int** ownerId;  
 **private** String ownerName;  
 **private** List<Pet> pets;  
   
 *//generate required constructors and setter/getter methods*}

**public class** Pet {  
 **private int** petId;  
 **private** String petName;  
 **private** String petType;  
 *//generate constructors and setter getters*}

a. Create an application and populate 10 Pet Objects in a HashMap and Write methods to

* 1. Add a pet to an Owner
  2. Remove a pet from an owner
  3. List all pets from an owner
  4. Find a pet by id