Java Programming

3-3: Collections – Part 2

Practice Activities

Lesson Objectives:

- Implement a HashMap
- Implement a stack by using a deque
- Define a link list
- Define a queue
- Implement a comparable interface

Vocabulary:

Identify the vocabulary word for each definition below.

Deque	A double-ended queue; a queue that can add and remove elements to the front or back of the list.
double-ended	The links of a LinkedList.
Collection	An interface used to define a group of objects. This includes lists and sets.
Map keys are always unique.	Maps that link a Key to a Value and may have duplicate Keys but cannot have duplicate Values.
treeSet	A list of elements that is dynamically stored.
Stack	A list of elements with a first in first out ordering.

Try It/Solve It:

1. What is the difference between a Queue and a Stack? Give an example of each.

   Queue - LIFO, Stack - FIFO

2. Below is a user implementation of a Stack using an ArrayList, create its implementation!
   a. Create a project named genericstack
   b. Create a class named GenericStack that uses the generic T as its parameter.
      - It has 2 local fields the ArrayList of T named items and an int variable top that keeps track of the top element in the list.
   c. Create a single constructor that sets top to zero.
   d. Create the following methods:
      - push - adds an item to the Stack

- pop - removes an item from the stack
  - if pop attempts to remove from an empty stack then an inline class named GenericStackException that implements RuntimeException should be called to display the message Underflow Error to the console.
- isEmpty return a Boolean value of true if the Stack is empty (top is zero).

e. Include a main method that will add 1, 2, 3 and 4 to the stack and then attempt 5 pops. Each pop should be displayed to screen.

3. Is it possible to add nodes to the beginning of a LinkedList? If so, how? What about adding a node to the end of a LinkedList? If this can be done, what method would be used?

```
private List<Integer> items = new ArrayList<>();
items.add(0, 123);
items.add(321);
```

4. What is the purpose of implementing the Comparable interface in one of our classes?

Override Sort Method

5. You are going to use a collection to store courses and their codes. Using the most appropriate collection store the following information.

Code	Course
CIT	Computing and Information Technology
CHI	Childcare and Early Education
MVS	Motor Vehicle Systems
BTH	Beauty Therapy
GDE	Graphic Design

```
private Map<String,String> course = new HashMap<>();
```

a. Print out the list of courses.

```
for (HashMap.Entry<String,String> pair : course.entrySet()){
        System.out.printf("Code: %-8s - Course: %-8s\n", pair.getKey(), pair.getValue());
}
```

b. Use the get method on one of the course codes to get the course name.

```
course.get("CHI")
```