

## **CE00527-5 Further Object Oriented Programming Tutorial 7**

### **Part 1 Writing a Generic class**

1. Write a class to represent a simple stack. Your class should be generic and implement only the methods below (ie do not implement the java Stack interface)

push(element: T): boolean     pushes an element onto the top of the stack  
pop(): T     returns and removes the top element of the stack  
              returns null if the stack is empty  
peek(): T     returns, but does not remove, the top element of the stack  
              returns null if the stack is empty

Use an ArrayList to hold the stack elements. Test your class by writing a simple driver class to create and use a stack of Strings or other objects.

### **Part 2 Using Wrapper classes**

2. Write a main method which
  - asks the user to input 5 doubles
  - stores the numbers in an ArrayList of type <Double>
  - prints the numbers out in reverse orderYour answer should take advantage of autoboxing/unboxing using a wrapper class.
- 3 Given the following main method, write a single static method (not a set of overloaded methods) which can be called successfully by all the methods in main.

```
public static void main(String[] args) {  
    Double answer1 = add(2, 7);  
    Number answer2 = add(new Integer(4), new Double(5.2));  
    double answer3 = add(8, 1.3);  
    System.out.println(answer1 + " " + answer2 + " " + answer3);  
}
```

### **Part 3 Using Navigable and BackedCollection methods**

- 4 A diary system consists of the following dates and appointments:

Date	Appointment
20 January 2012	appointment1
28 January 2012	appointment2
15 February 2012	appointment3
26 February 2012	appointment4

Create a TreeMap consisting of a Date type for the key and a String type for the value. Add the data in the above table.

Print out the diary system (i.e. dates and appointments)

Use a Navigable method to return and print out the date and appointment for the first appointment in January 2012 i.e you should input the key 20 January 2012

Use a Navigable method to return and print out the dates and appointments of all the appointments in February 2012.

Add a further appointment appointment8 on 20 February 2012 to your original TreeMap and again print out the dates and appointments of all the appointments in February 2012.

Hint: The following code allows you to change a string date to a Date date:

```
DateFormat dateformat = DateFormat.getDateInstance(DateFormat.LONG);
```

```
Date mydate=dateformat.parse("20 January 2012");
```

Hint: The method keySet() returns the set of keys for a TreeMap.

**You should put in your portfolio:**

- **a class diagram for your class which represents a generic stack**
- **the code listing for your class which represents a generic stack**
- **the methods you wrote in Part 2 questions 2 and 3**
- **the code listing for your class for part 3**
- **evidence (listing of your test methods with output) of testing of all your work**