Practical Exercise 3 – Generics

Overall Objective

To understand and write programs that use generic classes and interfaces.

Background

You will need to know:

1. basic Java programming knowledge
2. classes and interfaces
3. ArrayList
4. **generics**

Description

Part 1: Discussion

1. Are there any compile errors in (a) and (b)?

```
ArrayList dates = new
ArrayList();

dates.add(new Date());
dates.add(new String());
```

(a) Prior to JDK1.5

```
ArrayList <Date> dates =
     new ArrayList <Date>();
dates.add(new Date());
dates.add(new String());
```

(b) Since JDK 1.5

2. What is wrong in (a)? Is the code in (b) correct?

```
ArrayList dates = new
ArrayList();

dates.add(new Date());
Date date = dates.get(0);
```

(a) Prior to JDK1.5

```
ArrayList <Date> dates =
     new ArrayList <Date>();
dates.add(new Date());
Date date = dates.get(0);
```

(b) Since JDK 1.5

3. What are the benefits of using generic types?

4. How do you declare a generic type in a class?

5. How do you declare a generic method? How do you invoke a generic method?

6. What is raw type? Why is a raw type unsafe?

7. Refer to the *WildCardNeedDemo.java*, *AnyWildCardDemo.java* and *SuperWildCardDemo.java* programs, identify the differences among an *unbounded wildcard*, a *bounded wildcard*, and a *lower-bound wildcard.*

8. What happens if lines 14-15 in *SuperWildCardDemo.java* program are changed to:

```
public static <T> void add(GenericStack<T> stack1,
                           GenericStack<T> stack2 )
```

9. What is erasure?

10. If your program uses `ArrayList<String>` and `ArrayList<Date>`, does the JVM load both of them?

11. What is the problem with this code: `E[] elements = new E[capacity]`?

12. Can a method that uses a generic class parameter be static? Why?

Part 2: Programming Exercise

1. ***Minimum element in an* `ArrayList`**

   Implement the following method that returns the minimum element in an `ArrayList`.

```
public static <E extends Comparable<E>> E min(ArrayList<E> list)
```

2. ***Largest element in an* `ArrayList`**

   Implement the following method that returns the largest element in an `ArrayList`.

```
public static <E extends Comparable<E>> E max(ArrayList<E> list)
```