Java Programming Digital Assignment 1

Name: Om Ashish Mishra

Registration Number: 16BCE0789

Slot: G2

The Questions:

1. Explain in detail how Java is being platform independent.

The Answer:

A program is written in a language which is a human readable language. It may contain words, phrases etc which the machine does not understand. For the source code to be understood by the machine, it needs to be in a language understood by machines, typically a machine-level language. So, here comes the role of a compiler. The compiler converts the high-level language (human language) into a format understood by the machines. Therefore, a compiler is a program that translates the source code for another program from a programming language into executable code. This executable code may be a sequence of machine instructions that can be executed by the CPU directly, or it may be an intermediate representation that is interpreted by a virtual machine. This intermediate representation in Java is the Java Byte Code. Step by step Execution of Java Program: Whenever, a program is written in JAVA, the javac compiles it. The result of the JAVA compiler is the .class file or the bytecode and not the machine native code (unlike C compiler). The bytecode generated is a non-executable code and needs an interpreter to execute on a machine. This interpreter is the JVM and thus the Bytecode is executed by the JVM. And finally program runs to give the desired output.

1. Elaborate on how Garbage collection is happening in Java.

The Answer:

**Important points about Garbage Collection in Java**

1) Objects are created on the heap in Java irrespective of their scope e.g. local or member variable. while it's worth noting that class variables or static members are created in method area of [Java memory space](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html) and both heap and method area is shared between different thread.  
  
2) Garbage collection is a mechanism provided by Java Virtual Machine to reclaim heap space from objects which are eligible for Garbage collection.  
  
3) Garbage collection relieves Java programmer from memory management which is an essential part of C++ programming and gives more time to focus on business logic.  
  
4) Garbage Collection in Java is carried by a daemon thread called Garbage Collector.  
  
5) Before removing an object from memory garbage collection thread invokes [finalize() method](http://javarevisited.blogspot.com/2012/03/finalize-method-in-java-tutorial.html) of that object and gives an opportunity to perform any sort of cleanup required.  
  
6) You as Java programmer can not force garbage collection in Java; it will only trigger if JVM thinks it needs a garbage collection based on Java heap size.  
  
7) There are methods like System.gc() and Runtime.gc() which is used to send request of Garbage collection to JVM but it’s not guaranteed that garbage collection will happen.  
  
8) If there is no memory space for creating a new object in Heap Java Virtual Machine throws OutOfMemoryError or [java.lang.OutOfMemoryError heap space](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html)  
  
9) J2SE 5(Java 2 Standard Edition) adds a new feature called Ergonomics goal of ergonomics is to provide good performance from the JVM with a minimum of command line tuning. See [Java Performance The Definitive Guide](http://www.amazon.com/Java-Performance-The-Definitive-Guide/dp/1449358454?tag=javamysqlanta-20) for more details on garbage collection tuning.

3. Discuss how interactive event handling is done with Java Applets.

The Answer:

An applet is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal. Four methods in the Applet class gives you the framework on which you build any serious applet − init, start, stop, destroy

Change in the state of an object is known as event i.e. event describes the change in state of source. Events are generated as result of user interaction with the graphical user interface components. For example, clicking on a button, moving the mouse, entering a character through keyboard,selecting an item from list, scrolling the page are the activities that causes an event to happen.

The events can be broadly classified into Foreground and Background events.

Event Handling is the mechanism that controls the event and decides what should happen if an event occurs. This mechanism have the code which is known as event handler that is executed when an event occurs. Java Uses the Delegation Event Model to handle the events. This model defines the standard mechanism to generate and handle the events.Let's have a brief introduction to this model.

The Delegation Event Model has the following key participants namely:

* Source - The source is an object on which event occurs. Source is responsible for providing information of the occurred event to it's handler. Java provide as with classes for source object.
* Listener - It is also known as event handler.Listener is responsible for generating response to an event. From java implementation point of view the listener is also an object. Listener waits until it receives an event. Once the event is received , the listener process the event and then returns.

The benefit of this approach is that the user interface logic is completely separated from the logic that generates the event. The user interface element is able to delegate the processing of an event to the separate piece of code. In this model, Listener needs to be registered with the source object so that the listener can receive the event notification. This is an efficient way of handling the event because the event notifications are sent only to those listener that want to receive them.

Applets inherit a group of event-handling methods from the Container class. The Container class defines several methods, such as processKeyEvent and processMouseEvent, for handling particular types of events, and then one catch-all method called processEvent.

In order to react to an event, an applet must override the appropriate event-specific method.

The Abstract Window Toolkit (AWT) uses event driven programming to do processing of user actions, one that underlies all modern window systems programming. Within the AWT, all user actions belong to an abstract set of things called events. An event describes, in sufficient detail, a particular user action. Rather than the program actively collecting user-generated events, the Java run time notifies the program when an interesting event occurs. Programs that handle user interaction in this fashion are said to be event driven.

Event Handling provides four types of classes; they are:

1. Event Adapters
2. Event classes
3. Event Sources
4. Event Listeners

The following is a sample of event handling within the same class:

import java.awt.event.\*;

import java.awt.\*;

class EventActEx1 extends Frame implements ActionListener

{

TextField txtfld;

EventActEx1()

{

txtfld= new TextField();

txtfld.setBounds(65,60,190,20);

Button bt=new Button("Click me");

bt.setBounds(100,120,80,30);

bt.addActionListener(this);

add(bt);add(txtfld);

setSize(350,350);

setLayout(null);

setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

txtfld.setText("welcome 2 c-sharpcorner.com");

}

public static void main(String args[])

{

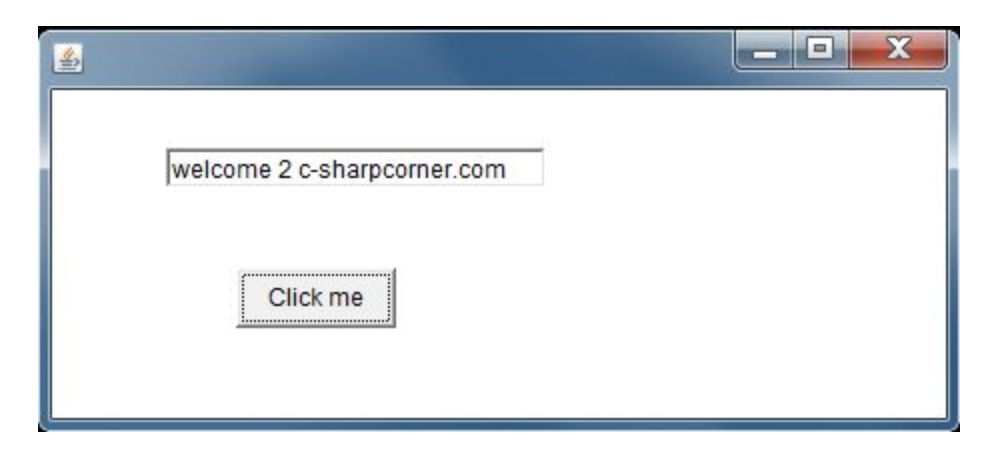
new EventActEx1();

}

}



On clicking Click me we get the following output:



This is how events are handled in Java using applets.

1. Enumerate the classes and interfaces in Java Collections with their inheritance hierarchy and explain their usefulness with method signatures.

The Answer:

A Java collection framework provides an architecture to store and manipulate a group of objects. A Java collection framework includes the following:

* Interfaces
* Classes
* Algorithm

The JAVA Collections Interface constitutes of the Iteration Interface

**Iterator interface**: Iterator is an interface that iterates the elements. It is used to traverse the list and modify the elements. Iterator interface has three methods which are mentioned below:

1. **public boolean hasNext()** – This method returns true if the iterator has more elements.
2. **public object next()** – It returns the element and moves the cursor pointer to the next element.
3. **public void remove()** – This method removes the last elements returned by the iterator.

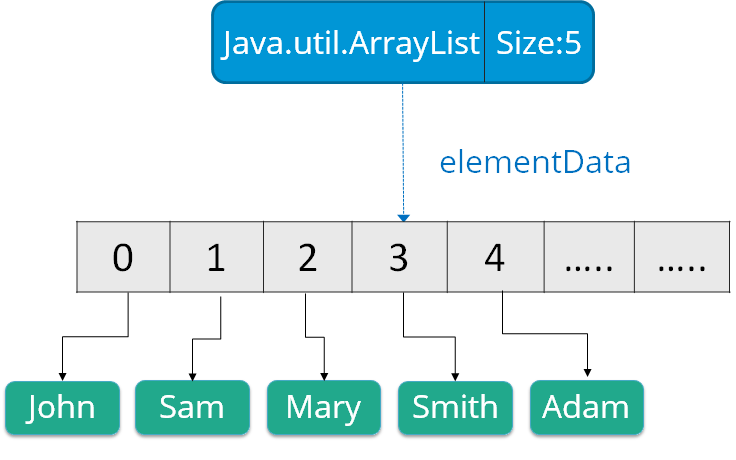
There are three components that extend the collection interface i.e List, Queue and Sets.

## Java collections: List

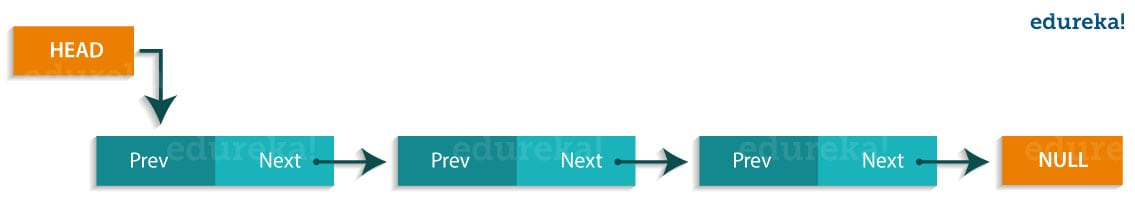
A List is an ordered Collection of elements which may contain duplicates. It is an interface that extends the Collection interface. Lists are further classified into the following:

1. ArrayList
2. LinkedList
3. Vectors

**Array list:** ArrayList is the implementation of List Interface where the elements can be dynamically added or removed from the list. Also, the size of the list is increased dynamically if the elements are added more than the initial size.

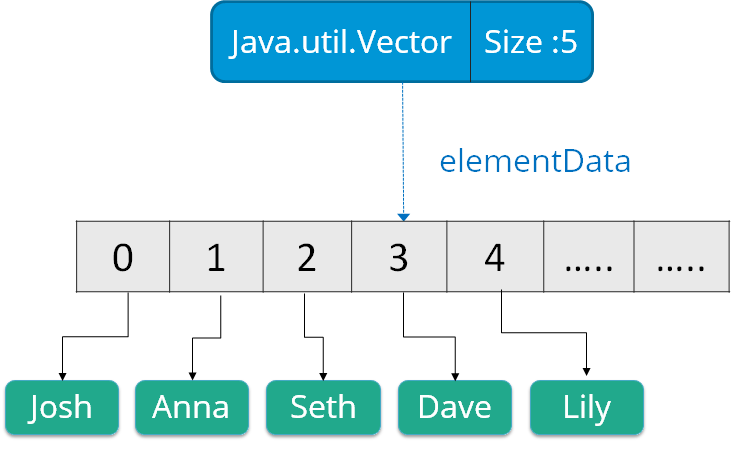


**Linked List**: Linked List is a sequence of links which contains items. Each link contains a connection to another link.



**Vectors** : Vectors are similar to arrays, where the elements of the vector object can be accessed via an index into the vector. Vector implements a dynamic array. Also, the vector is not limited to a specific size, it can shrink or grow automatically whenever required. It is similar to ArrayList, but with two differences :

* Vector is synchronized.
* Vector contains many legacy methods that are not part of the collections framework.

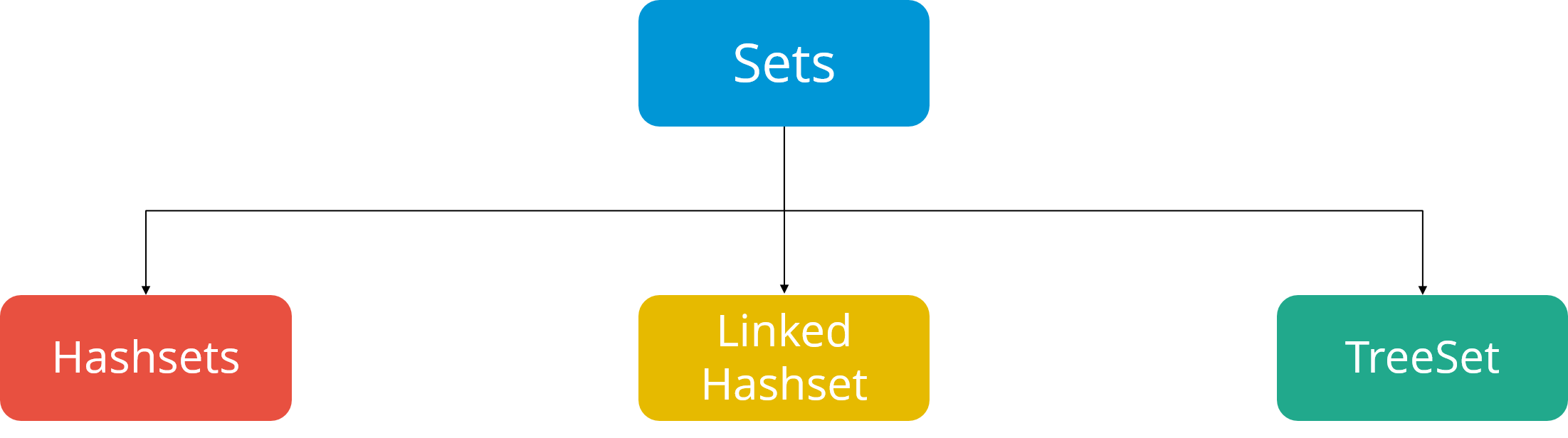


## Java collections: Queue

Queue in Java follows a FIFO approach i.e. it orders the elements in First In First Out manner. In a queue, the first element is removed first and last element is removed in the end. Each basic method exists in two forms: one throws an exception if the operation fails, the other returns a special value.

## Java Collections: Sets

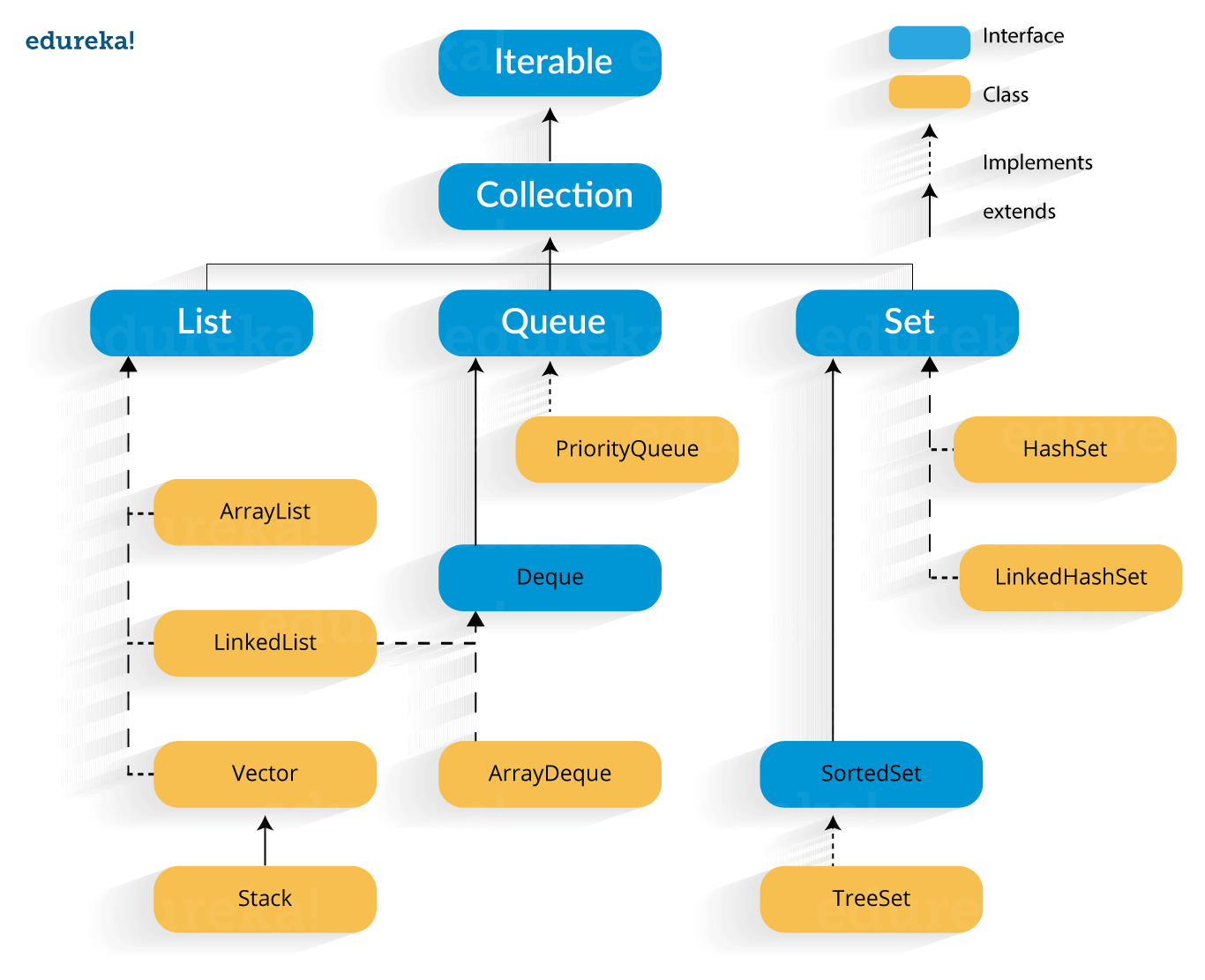
A Set refers to a collection that cannot contain duplicate elements. It is mainly used to model the mathematical set abstraction. Set has its implementation in various classes such as HashSet, TreeSetand LinkedHashSet.



**HashSet**: Java HashSet class creates a collection that use a hash table for storage. Hashset only contain unique elements and it inherits the AbstractSet class and implements Set interface. Also, it uses a mechanism *hashing* to store the elements.

**Linked Hashset** Java LinkedHashSet class is a Hash table and Linked list implementation of the set interface. It contains only unique elements like HashSet. Linked HashSet also provides all optional set operations and maintains insertion order.  
**TreeSet** TreeSet class implements the Set interface that uses a tree for storage. The objects of this class are stored in the ascending order. Also, it inherits AbstractSet class and implements NavigableSet interface. It contains only unique elements like HashSet. In TreeSet class, access and retrieval time are faster.

The inheritance hierarchy of the classes and interfaces is given as:



1. List down the similarities and differences between ASP and JSP.

The Answer:

The Similarities:

JSP and ASP are both server side scripting languages. JSP (Java Server Pages) and ASP (Active Server Pages) are two of the commonly used server side scripting languages that are used today in web development.  
  
The differences:

1. JSP and ASP are both server side scripting languages

2. JSP is from Sun Microsystems while ASP is from Microsoft

3. ASP costs money while JSP is free.

4. ASP code is interpreted while JSP code is compiled at run time

5. JSP code can run faster than ASP if there are fewer changes

6. Majority of Windows users use ASP while users of open source operating systems like Linux use JSP among others.

|  |  |
| --- | --- |
| **JSP** | **ASP** |
| JSP is Java Server Pages. | ASP is Active Server Pages. |
| JSP is from Sun Microsystems, recently acquired by Oracle. | ASP is from Microsoft. |
| JSP is free. | ASP costs money. |
| JSP code is compiled at run time. | ASP code is interpreted. |
| JSP code can run faster than ASP if there are fewer changes. | In this case, ASP is little slower. |
| Users of open source operating systems like Linux use JSP among others. | Majority of Windows users use ASP. |
| Maker & Programming Language: On the other hand jsp technology is not a Microsoft made. It uses Java language as the programming language. and accordingly if you know Java then you can easily start creating your website in JSP. | Maker & Programming Language: ASP uses Visual Basic language. In short if you know how to program in Visual Basic then you can easily make ASP pages. Additionally the new generation of ASP is called. ASP .NET which can be programmed in several programming languages. mostly on VB . NET and C#. |
| Database: jsp is not known to be twined with any database. so you can connect it to any database system simply by using the needed the driver (libraries) for the database then connect. I use MySQL my self when coding in jsp. | Database: Theoretically you can connect to any database through ADO if you've used the required addon. though asp fully supports connecting to access database (no server required) and support also MS SQL connection. |
| JSP website is mostly supported by Apache Tomcat web server and work mostly on Linux based web server. But it also runs on JBOSS and IBM application servers. JSP is definitely best if you're running a linux or UNIX server, and should work more reliability over all. It can take a bit of work to configure tomcat to work with Apache though. | Mostly it works on Microsoft IIS server. other servers can support asp also but mostly it works on IIS web server on windows machine. |