Assignment-7

Page No.:

Date:

Q.1] What is Text I/O ?

Ans Text I/o means text input-output. Data stored in a text file are represented in humen-readable form. Java programs are stored in text file and can read by a text editor. This text i/o consists of sequence of characters and numbers etc., Hence, it is known as Text I/O.

Q.2] What is Binary I/o ?

Ans Binary I/o means Binary Input-Output. Data stored in a binary file are represented in binary form. You cannot read binary files. This files are designed to be read by programs. For example, Java classes are stored in binary files and are read by the JVM. The advantage of binary files is that they are more efficient to process than text files. Therefore, it is known as Binary I/o.

Q.3] What is Random Access File?

Ans All of the streams you have used so far are known as read-only or write-only streams.

- The external files that cannot be updated

Page No.:
Date:

- The external files of these streams are sequential files that cannot be updated without creating a new file.
- It is often necessary to modify files or to insert new records into files.
- Java provides the RandomAccessFile class to allow a file to be read from and write to at random locations.

Q.4] What is Tail Recursion?

Ans A recursive method is said to be tail recursive if there are no pending operations to be performed on return from a recursive call.

For example:-

```
void print (int n)
{
    if (n<0) return;
    count << "   " << n;
    print (n-1);
}
```

- So, it is known as Tail Recursion.

Q.5] Define Generics Class.

Ans Generics is the capability to parameterize types. With this capability, you can define a class or a method with generic types that can be substituded using concrete types by the compiler.

- For example, you man define a generic stack class that stores the elements of a generic type.

- From this generic ~~types~~ class, you may create a stack object for holding strings and a stack object for holding numbers.

- Here, strings and numbers are concrete types that replace the generic type.

Q.6] Explain Lists

Ans List Interface is the subinterface of collection.

- It contains index-based methods to insert and delete elements.

- It is a factory of listIterator interface.

- List is an ordered collection of objects in which duplicate values can be stored.

- Since list preserves the insertion order, it allows positional access and insertion of elements.

- List Interface is implemented by the classes of ArrayList, LinkedList, vector and stack

Q.7] Explain Comparator Interface.

Ans Java Comparator interface is used to order the objects of a user-defined class.

- This interface is found in java.util package and contains 2 methods compare (Object obj1, Object obj2) and equals (Object element)

- It provides multiple sorting sequences, i.e., you can sort the elements on the basis of any data member, for example, rollno, name, age or anything else.

- Sometimes you want to insert elements of different types into a tree set. The elements may not be instances of comparable or are not comparable

- You can define a comparator to compare these elements. To do so, create a class that implements the java.util.comparator interface.

- The comparator interface has two methods, compare and equals.

Q8) Explain Vector.

ans Vector is the same as ArrayList, except that vector contains the synchronized methods for accessing and modifing the vector.

- None of the new collection data structures introduced so far are synchronized.

- If synchronization is required, ~~of~~ you can use the synchronized versions of the collection classes.

- This classes are introduced later in the section, " The collection Class"

- Vector ~~is~~ ~~a~~ implements a dynamic Array. It proves to be very useful if you don't know the size of the array in advance or you just need one that can change sizes over the lifetime of a program

Page No.:

Date:

Q9] Explain Queue and Priority Queue:

Ans A queue is a first-in / first-Out data structure. Elements are appended to the end of the queue and are removed from the beginning of the queue.

- In a priority queue, elements are assigned priorities. When accessing elements, the elements with the highest priority is removed first.