1. **What does the “static” keyword mean ?**

**Ans:** Static is a modifier for variables and functions inside a class.  It, like any other modifier, is placed before the object whose instantiation is modified.

1. **Can you override private or static method in Java ?**

**Ans: No we cannot override private or static method in java .**

1. **Can you access non static variable in static context ?**

**Ans: We cannot access non static variable in static context.**

1. **What are the Data Types supported by Java ?**

**Ans: Byte-8bit ,short-16bit,int-32bit,long-64bit,float=32bit,double-64bit,char-16bit,Boolean-true/false.**

1. **What is Autoboxing and Unboxing ?**

**Ans: Boxing- converting a value type to reference type is called boxing.**

**AutoBoxing- if it is getting implemented automatically then it is called as an autoboxing.**

**Unboxing-converting reference type to value value type is called unboxing.**

1. **Difference between an Interface and an Abstract class ?**

**Ans: Abstract class:**

* Abstract class can **have abstract and non-abstract** methods.
* Abstract class **can have final, non-final, static and non-static variables**.
* Abstract class **can have static methods, main method and constructor**.

**Interface :**

* Interface can have **only abstract** methods.
* Interface **supports multiple inheritance**.
* Interface **can't have static methods, main method or constructor**.

1. **What is the difference between pass by reference and pass by value ?**

**Ans: Pass By Value:** Actual parameter expressions that are passed to a method are evaluated and a value is derived. Then this value is stored in a location and then it becomes the formal parameter to the invoked method. This mechanism is called pass by value

Pass By Reference: he formal parameter is just an alias to the actual parameter. It refers to the actual argument. Any changes done to the formal argument will reflect in actual argument and vice versa.

1. **What are the basic interfaces of Java Collections Framework ?**

**Ans:** boolean add(Object obj), boolean addAll(Collection c), void clear( ), boolean equals(Object obj), int hashCode( ), boolean isEmpty( ), Iterator iterator( ), int size( ), boolean retainAll(Collection c), Object[ ] toArray(Object array[ ]), Object[ ] toArray( ) etc.

1. **Why do we use an Iterator while using Java Collections ?**

**Ans:** Iterator enables you to cycle through a collection, obtaining or removing elements. ListIterator extends Iterator to allow bidirectional traversal of a list, and the modification of elements.

1. **What is the difference betweenHashMap and Hashtable ?**

**Ans: HashMap**

* HashMap is **non synchronized**. It is not-thread safe and can't be shared between many threads without proper synchronization code.
* HashMap **allows one null key and multiple null values**.
* HashMap is **fast**.

**Hashtable:**

* Hashtable is **synchronized**. It is thread-safe and can be shared with many threads.
* Hashtable **doesn't allow any null key or value**.
* Hashtable is **slow**.

1. **What is difference between Array and ArrayList ? When will you use Array over ArrayList ?**

**Ans: Array:**

**>** An array is basic functionality provided by Java.

> Array is a fixed size data structure.

>Array can contain both primitive data types as well as objects of a class depending on the definition of the array.

**ArrayList**

**>** ArrayList is part of collection framework in Java.

> Dynamic sized arrays in Java that implement List interface.

> ArrayList only supports object entries, not the primitive data types.

1. **When to use parseInt method?**

**Ans:** The Integer.parseInt() java method is used primarily in parsing a String method argument into an Integer object. The Integer object is a wrapper class for the int primitive data type of java

1. **What is the difference between StringBuffer and StringBuilder class?**

**Ans: StringBuffer:**

**>** StringBuffer is synchronized i.e. thread safe. It means two threads can't call the methods of StringBuffer simultaneously.

> StringBuffer is less efficient than StringBuilder.

StringBuilder:

> StringBuilder is non-synchronized i.e. not thread safe. It means two threads can call the methods of StringBuilder simultaneously.

> StringBuilder is more efficient than StringBuffer.

1. **What is finalize method?**

**Ans: It** is called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

1. **What is the difference between final, finally and finalize method?**

**Ans: Final: final is a keyword.**

**Finally: Finally is a block.**

**Finalize: Finalize is a method.**

1. **What is the difference between throw and throws keyword?**

**Ans: Throw :**

**>** throw keyword is used to explicitly throw an exception.

> throw is followed by an instance.

. Throws:

> throws keyword is used to declare an exception.

> throws is followed by class.

1. **When and why to use “super” keyword ?**

**Ans:** super is used to refer immediate parent class instance variable.

super() is used to invoke immediate parent class constructor

super is used to invoke immediate parent class method.

1. **What is “jar” file and how to create and use it?**

**Ans:**  tool of JDK provides the facility to create the executable jar file. An executable jar file calls the main method of the class...

We can create jar file by using :main-class:First

1. **What is the purpose of fileInputStream and fileReader class?**

**Ans:** The difference between fileinputestream and filereader  is, fileinputstream  reads the file byte by byte and filereader reads the file character by character

.

1. **What is the purpose of fileOutputStream and fileWriter class?**

**Ans:** FileOutputStream is an OutputStream for writing bytes to a file.  OutputStreams do not accept chars by wrapping it in an OutputStreamWriter. FileWriter is a Writer that talks to files FileWriter is the natural class for use with Unicode Strings.

All the programs should be written in Java Language.

1. Write a program to implement Binary Search?

Ans: public static void main(String args[])

{

int n, i, search, first, last, middle;

int arr[] = new int[50];

Scanner scan = new Scanner(System.in);

System.out.print("Enter Total Number of Elements : ");

n = scan.nextInt();

System.out.print("Enter " +n+ " Elements : ");

for(i=0; i<n; i++)

{

arr[i] = scan.nextInt();

}

System.out.print("Enter a Number to Search..");

search = scan.nextInt();

first = 0;

last = n-1;

middle = (first+last)/2;

while(first <= last)

{

if(arr[middle] < search)

{

first = middle+1;

}

else if(arr[middle] == search)

{

System.out.print(search+ " Found at Location " +middle);

break;

}

else

{

last = middle - 1;

}

middle = (first+last)/2;

}

if(first > last)

{

System.out.print("Not Found..!! " +search+ " is not Present in the List.");

}

}

1. Write a program to implement Bubble sort?

Ans: public static void main(String args[])

{

int n, i, j, temp;

int arr[] = new int[50];

Scanner scan = new Scanner(System.in);

System.out.print("Enter Total Number of Elements : ");

n = scan.nextInt();

System.out.print("Enter " +n+ " Numbers : ");

for(i=0; i<n; i++)

{

arr[i] = scan.nextInt();

}

System.out.print("Sorting Array using Bubble Sort Technique...\n");

for(i=0; i<(n-1); i++)

{

for(j=0; j<(n-i-1); j++)

{

if(arr[j] > arr[j+1])

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

System.out.print("Array Sorted Successfully..!!\n");

System.out.print("Sorted List in Ascending Order : \n");

for(i=0; i<n; i++)

{

System.out.print(arr[i]+ " ");

}

}

1. Write a program to implement the stack operations?

Ans: static void showpush(Stack st, int a) {   
st.push(new Integer(a));   
System.out.println("push(" + a + ")");   
System.out.println("stack: " + st);   
}   
static void showpop(Stack st) {   
System.out.print("pop -> ");   
Integer a = (Integer) st.pop();   
System.out.println(a);   
System.out.println("stack: " + st);   
}   
public static void main(String args[]) {   
Stack st = new Stack();   
System.out.println("stack: " + st);   
showpush(st, 42);   
showpush(st, 66);   
showpush(st, 99);   
showpop(st);   
showpop(st);   
showpop(st);   
try {   
showpop(st);   
} catch (EmptyStackException e) {   
System.out.println("empty stack");   
}   
}   
}

1. Write a program to search any data in given LinkList without using collections framework?

Ans: public static void main(String[] args)

{

int[] data = { 10,20,30,15,40,66};

Object[][] linkedList = new Object[1][2];

createLinkedList(data,linkedList);

printLinkedList(linkedList);

}

public static void createLinkedList(int []data,Object[][] linkedList)

{

Object node[][] = null;

Object newnode[][] = null;

node=new Object[1][2];

linkedList[0][1]=node;

for(int i=0;i<data.length;i++)

{

node[0][0]=new Integer(data[i]);

newnode=new Object[1][2];

node[0][1]=newnode;

node=newnode;

}

node =null;

newnode = null;

    }

public static void printLinkedList(Object[][] linkedList)

{

Object node[][] = null;

node=linkedList;

node =(Object [][]) node[0][1];

while(node[0][1]!= null)

{

System.out.print("  "+node[0][0]+" --->" );

node=(Object [][]) node[0][1]

}

System.out.print("null" );

}

1. Write a program to Add and Delete data inside LinkList without using collections framework?

Ans: public class LinkedListDemo {

public static void main(String args[]) {

LinkedList ll = new LinkedList();

ll.add("F");

ll.add("B");

ll.add("D");

ll.add("E");

ll.add("C");

ll.addLast("Z");

ll.addFirst("A");

ll.add(1, "A2");

System.out.println("Original contents of ll: " + ll);

ll.remove("F");

ll.remove(2);

System.out.println("Contents of ll after deletion: " + ll);

ll.removeFirst();

ll.removeLast();

System.out.println("ll after deleting first and last: " + ll);

Object val = ll.get(2);

ll.set(2, (String) val + " Changed");

System.out.println("ll after change: " + ll);

}