List Interface Exercises

Interface:

1) What does the following code output?

```
public class Main
{
	public static void main(String[] args)
	{
	List<Integer> aList = new ArrayList<Integer>(Arrays.asList(1,2,3));
	List<Integer> aList2 = new ArrayList<Integer>(Arrays.asList(4,5,6));
	List<Integer> lList = new LinkedList<Integer>();

	lList.addAll(aList2);
	lList.addAll(aList);

	System.out.println(lList);
	}
}
```

2) Does the following compile?

```
public class Main
{
        public static void main(String[] args)
        {
        List<Integer> aList = new ArrayList<Integer>(Arrays.asList(1,2,3));
        List<Integer> aList2 = new ArrayList<Integer>(Arrays.asList(4,5,6));
        List<Float> lList = new LinkedList<Float>();

        lList.addAll(aList2);
        lList.addAll(aList);

        System.out.println(lList);
        }
}
```

3) What is the output of the following?

```
public class Main
{
        public static void main(String[] args)
        {
        List<Integer> aList = new ArrayList<Integer>(Arrays.asList(1,2,3));
        List<Integer> aList2 = new ArrayList<Integer>(Arrays.asList(4,5,6));
        List<Integer> lList = new LinkedList<Integer>();
        lList.addAll(aList2);
        lList.addAll(aList);
        lList.clear();
        System.out.println(lList);
        }
}
```

4) What is the output of the following?

```
import java.util.*;


public class Main
{
        public static void main(String[] args)
        {
        List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));
        if(lList.contains(4))
        {
                System.out.println("lList contains 4");
        }
        else
        {
                System.out.println("lList does not contain 4");
        }
        }
}
```

5) Does the following compile?

```
List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));
if(lList.contains(4.0))
{
        System.out.println("lList contains 4");
}
else
{
        System.out.println("lList does not contain 4");
}
```

6) What is the output of the following?

```
List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));

if(lList.contains(4.0))

{

	System.out.println("lList contains 4");

}

else

{

	System.out.println("lList does not contain 4");

}
```

10) Does the following compile?

```
List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));

List<Float> oList = new
LinkedList<Float>(Arrays.asList(1.0f,2.0f,3.0f,4.0f,5.0f,6.0f,7.0f));

if(lList.containsAll(oList))

{

	System.out.println("lList contains 4");

}

else

{

	System.out.println("lList does not contain 4");

}
```

11) What is the output of the following?

```
List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));

List<Integer> vList = new Vector<Integer>(Arrays.asList(1,2,3,4,5,6));

if(lList.equals(vList))

{

      System.out.println("lList is equal to vList");

}

else

{

      System.out.println("lList is equal to vList");

}
```

12) What is the output of the following?

```
List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));

System.out.println(lList.indexOf(1.0f));
```

13) What is the output of the following?

```
List<Integer> lList = new LinkedList<Integer>(Arrays.asList(1,2,3,4,5,6));

System.out.println(lList.indexOf(2));
```

14) What is the output of the following?

```
public class Foo
{
        public int bar;

        public Foo(int bar_)
        {
        bar = bar_;
        }
}
```

```
public static void main(String[] args)
{
        List<Foo> fooList = new ArrayList<Foo>(Arrays.asList(new Foo(1), new Foo(2)));
        List<Foo> fooVector = new Vector<Foo>(Arrays.asList(new Foo(1), new Foo(2)));
        if(fooList.equals(fooVector))
        {
        System.out.println("fooList is equal to fooVector");
        }
        else
        {
        System.out.println("fooList is not equal to fooVector");
        }
}
```

15) What is the output of the following?

```
List<Foo> fooList = new ArrayList<Foo>(Arrays.asList(new Foo(1), new Foo(2)));
List<Foo> fooVector = new Vector<Foo>(Arrays.asList(new Foo(1), new Foo(2)));
System.out.println(fooList.indexOf(new Foo(2)));
```

16) What is the output of the following?

```
List<Integer> aList = new ArrayList(Arrays.asList(1,2,3,4,5,3,6));
System.out.println(aList.lastIndexOf(3));
```

17) What is the output of the following?

```
List<Integer> aList = new ArrayList(Arrays.asList(1,2,3,4,5,3,6));
aList.removeAll(Arrays.asList(1,2,3));
System.out.println(aList);
```

18) What is the output of the following?

```
List<Integer> aList = new ArrayList(Arrays.asList(1,2,3,4,5,3,6));
if(!aList.removeAll(Arrays.asList(7,8,9,10)))
{
	System.out.println("No elements removed");
}
System.out.println(aList);
```

19) What is the output of the following?

```
List<Integer> aList = new ArrayList(Arrays.asList(1,2,3,4,5,3,6));
if(!aList.retainAll(Arrays.asList(1.0f,2.0f)))
{
	System.out.println("List not changed");
}
System.out.println(aList);
```

20) What is the output of the following?

```
List<Integer> aList = new ArrayList(Arrays.asList(1,2,3,4,5,3,6));

if(!aList.retainAll(Arrays.asList(1,2,3,4,5,6)))

{

	System.out.println("List not changed");

}

System.out.println(aList);

}
```

21) Write a function, public static int[] toArray(List<Integer> list), to convert a List<Integer> to an array of int. Use List.toArray.

22) What is the output of the following?

```
List<Integer> aList = new ArrayList(Arrays.asList(1,2,3,4,5,3,6));

System.out.println(aList.subList(2, aList.lastIndexOf(3)));
```

23) Write a function, public static void printList(List<String> list) that replicates the printing of System.out.println(list).

24) Will the following run without issue?

```
List<String> animals = new ArrayList(Arrays.asList("dog","cat","sheep"));

Iterator<String> iter = animals.iterator();

while(iter.hasNext())

{

	iter.next();

	iter.remove();

	iter.next();

	iter.remove();

}
```

25) Using ListIterator write a function, public static void printNumbersDigits(int number). This function should create a List containing the digits of the number as elements. Then using a ListIterator traverse the List backwards printing out the digits separated by commas. E.x. `1,2,3,4,5,6`

26) Write a function, public static void setAll(String name, String set, List<String> list), this function should replace all instances of name with set.

27) Should you prefer raw types to generics?

a) Yes

b) No

28) Which of the following is the correct way to get a value from a raw type List?

```
a) int a = (Integer)list.get(0);
b) int b = list<integer>.get(0);
c) int c = list.get(0);
d) int d = list.get(0) as Integer;
```

29) List is a?

a) Interface
b) Class
c) Object
d) Method

Answer: a

30) Which of the following implement List?

I) ArrayList
II) Vector
III) Array

a) I only
b) II only
c) I, II, and II
d) I and II

Answer: d

31) What is the output of the following?

```
List<Integer> list = new ArrayList(3);

list.add(new Integer(1));
list.add(new Integer(2));
list.add(new Integer(1));

Integer one = list.get(0);
Integer two = list.get(2);

if(one == two)
   System.out.println(one +  " is equql to " + two);
else
   System.out.println(one + " is NOT equal to " + two);
```

a) 1 is equal to 1
b) 2 is NOT equal to 1
c) 1 is NOT equal to 2
d) 1 is NOT equal to 1

Answer: d

32) What is the output of the following?

```
List<Integer> list = new ArrayList();
ListIterator<Integer> lIter = list.listIterator();
lIter.add(0);
lIter.add(1);
System.out.println(list);
```

a) No output
b) [0, 1]
c) [1, 0]
d) [1, 1]

Answer: b

33) What is the output of the following?

```
List<Integer> list = new ArrayList(Arrays.asList(1,2,1,3,1,4,1,5));
ListIterator<Integer> lIter = list.listIterator();

while(lIter.hasNext())
{
   if(lIter.next() == 1)
   {
         lIter.add(lIter.next() + 1);
   }
}
```

```
a) [1, 2, 3, 2, 3, 4, 1, 4, 5, 1, 5, 6]
b) [1, 2, 3, 2, 3, 4, 3, 4, 5, 4, 5, 6]
c) [1, 2, 3, 2, 3, 4, 1, 4, 5, 2, 5, 6]
d) No output, infinite loop
```

Answer: a