**Q1**

Whats is the output?  
  
import java.util.ArrayList;  
import java.util.List;  
public class Test {       
      public static void main(String[] args) {  
           List<String> list = new ArrayList<>();  
           list.add("English");  
           list.add("Hindi");  
           list.add("Kannada");  
           list.add("Marathi");  
           list.add("Tamil");  
           list.add("Telugu");  
           list.add(2,"Oriya");  
           System.out.print(list.get(4));  
           list.remove(3);  
           System.out.print(list.get(4));  
      }  
}

*Single Choice - Select one correct answer from the options list.*

* TamilTelugu
* KannadaMarathi
* MarathiTamil
* Exception at runtime

**Q2**

Refer the given code :  
  
public class Test {  
public static void main(String... args) {  
  
Set s = new TreeSet();  
 s.add("7");  
 s.add(9);  
 Iterator itr = s.iterator();  
 while (itr.hasNext())  
 System.out.print(itr.next() + " ");  
}  
}

*Single Choice - Select one correct answer from the options list.*

* Compile error
* Runtime Exception
* 7 9
* None of the above

**Q3**

Identify 2 benefits of using ArrayList over array in Software Development.

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

* reduces memory footprint
* implements Collection API
* is multithread safe
* dynamically resizes based on the number of elements in the list

**Q4**

 Given:  
  
public class Concert {  
static class PowerOutage extends Exception {}  
static class Thunderstorm extends Exception {}  
public static void main(String[] args) {  
try {  
new Concert().listen();  
System.out.println("a");  
} catch(PowerOutage | Thunderstorm e) {  
e = new PowerOutage();  
System.out.println("b");  
} finally { System.out.println("c"); }  
}  
public void listen() throws PowerOutage, Thunderstorm{ }  
}  
  
What will this code print?

*Single Choice - Select one correct answer from the options list.*

* a
* ab
* ac
* abc
* compilation error

**Q5**

If we do:  
  
ArrayList lst = new ArrayList();  
  
What is the initial capacity of the ArrayList lst ?

*Single Choice - Select one correct answer from the options list.*

* 10
* 8
* 15
* 12

**Q6**

Given:   
  
1. public static Collection get() {   
2.          Collection sorted = new LinkedList();   
3.          sorted.add("B"); sorted.add("C"); sorted.add("A");   
4.          return sorted;   
5. }   
6. public static void main(String[] args) {   
7.           Collection list = get();  
7.          for (Object obj: list) {   
8.                    System.out.print(obj + ", ");   
9.          }   
10. }   
  
What is the result?

*Single Choice - Select one correct answer from the options list.*

* A, B, C
* B, C, A
* Compilation fails.
* The code runs with no output.

**Q7**

Refer the code below:  
  
import java.util.ArrayList;  
 import java.util.List;  
 public class Test{   
       public static void main(String args[]) {  
        List<Integer> list = new ArrayList<Integer>();   
    list.add(0, 59);  
    int total = list.get(0);  
    System.out.println(total);    
     }  
}

*Single Choice - Select one correct answer from the options list.*

* Gives output : 0
* Will not compile
* Gives output : 59
* Runtime Exception

**Q8**

What is the result?  
  
1. import java.util.\*;  
2.  
3. public class LetterASort{  
4. public static void main(String[] args) {  
5. ArrayList<String> strings = new ArrayList<String>();  
6. strings.add("aAaA");  
7. strings.add("AaA");  
8. strings.add("aAa");  
9. strings.add("AAaa");  
10. Collections.sort(strings);  
11. for (String s : strings) { System.out.print(s + " "); }  
12. }  
13. }

*Single Choice - Select one correct answer from the options list.*

* Compilation fails.
* aAaA aAa AAaa AaA
* AAaa AaA aAa aAaA
* AaA AAaa aAaA aAa
* aAa AaA aAaA Aaaa

**Q9**

Which code, inserted at line 4, guarantees that this program will output [1, 2]?  
  
1. import java.util.\*;  
2. public class Example {  
3. public static void main(String[] args) {  
4. // insert code here  
5. set.add(new Integer(2));  
6. set.add(new Integer(1));  
7. System.out.println(set);  
8. }  
9. }

*Single Choice - Select one correct answer from the options list.*

* Set set = new TreeSet();
* Set set = new HashSet();
* Set set = new SortedSet();
* List set = new SortedList();
* Set set = new LinkedHashSet();

**Q10**

View the Exhibit.  
public class Hat {  
public int ID =0;  
public String name = "hat";  
public String size = "One Size Fit All";  
public String color="";  
public String getName() { return name; }  
public void setName(String name) {  
this.name = name;  
}  
}  
Given  
public class TestHat {  
public static void main(String[] args) {  
Hat blackCowboyHat = new Hat();  
}  
}  
  
Which statement sets the name of the Hat instance?

*Single Choice - Select one correct answer from the options list.*

* blackCowboyHat.setName = "Cowboy Hat";
* setName("Cowboy Hat");
* Hat.setName("Cowboy Hat");
* blackCowboyHat.setName("Cowboy Hat");

**Q11**

 Which of the following are true statements?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

* The Iterator interface declares only three methods: hasNext, next and remove.
* The ListIterator interface extends both the List and Iterator interfaces.
* The ListIterator interface provides forward and backward iteration capabilities.
* The ListIterator interface provides the ability to modify the List during iteration.
* The ListIterator interface provides the ability to determine its position in the List.

**Q12**

Which statements creates an ArrayList of Strings with an initial capacity of 20?   
(Choose all that apply)

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

* ArrayList<String> names = new ArrayList<>( );
* ArrayList<String> names = new ArrayList<>(20);
* ArrayList<String> names = new ArrayList<String>( );
* ArrayList<String> names = new ArrayList<String>(20);

**Q13**

Given:   
  
1. public static Collection get() {   
2.          Collection sorted = new LinkedList();   
3.          sorted.add("B"); sorted.add("C"); sorted.add("A");   
4.          return sorted;   
5. }   
6. public static void main(String[] args) {   
7.          for (Object obj: get()) {   
8.                    System.out.print(obj + ", ");   
9.          }   
10. }   
  
What is the result?

*Single Choice - Select one correct answer from the options list.*

* A,B,C
* B,C,A
* Compilation Fails
* The code runs without output
* Exception thrown

**Q14**

Given the code fragment:  
  
1. ArrayList<Integer> list = new ArrayList<>(1);  
2. list.add(1001);  
3. list.add(1002);  
4. System.out.println(list.get(list.size()));  
  
What is the result?

*Single Choice - Select one correct answer from the options list.*

* Compilation fails due to an error on line 1.
* An exception is thrown at run time due to error on line 3
* An exception is thrown at run time due to error on line 4
* 1002

**Q15**

What, inserted at line 39, will sort the keys in the props HashMap?  
  
34. HashMap props = new HashMap();   
35. props.put("key45", "some value");   
36. props.put("key12", "some other value");   
37. props.put("key39", "yet another value");   
38. Set s = props.keySet();   
39. // insert code here

*Single Choice - Select one correct answer from the options list.*

* Arrays.sort(s);
* s = new TreeSet(s);
* Collections.sort(s);
* s = new SortedSet(s);

**Q16**

Which line of code marks the earliest point that an object referenced by intObj becomes   
a candidate for garbage collection?  
  
11. public void genNumbers() {  
12. ArrayList numbers = new ArrayList();  
13. for (int i=0; i<10; i++) {  
14. int value = i \* ((int) Math.random());  
15. Integer intObj = new Integer(value);  
16. numbers.add(intObj);  
17. }  
18. System.out.println(numbers);  
19. }

*Single Choice - Select one correct answer from the options list.*

* Line 16
* Line 17
* Line 18
* Line 19
* The object is NOT a candidate for garbage collection.

**Q17**

Given the following code:  
  
package test;  
import java.util.ArrayList;  
public class Roller {  
public static void main(String[] args) {  
ArrayList<String> collector = new ArrayList<String>( );  
collector.add("Study");  
collector.add(Integer.toOctalString(Integer.MAX\_VALUE));  
ArrayList<String> names = new ArrayList<>( );  
names.addAll(collector);  
names.add("Study");  
names.add(null);  
System.out.println(names.size( ));  
}  
}  
  
What is printed on the console?

*Single Choice - Select one correct answer from the options list.*

* 0
* 2
* 3
* 4

**Q18**

Given a pre-generics implementation of a method:  
  
11. public static int sum(List list) {   
12. int sum = 0;   
13. for ( Iterator iter = list.iterator(); iter.hasNext(); ) {   
14. int i = ((Integer)iter.next()).intValue();   
15. sum += i;   
16. }   
17. return sum;   
18. }   
  
Which three changes must be made to the method sum to use generics? (Choose three.)

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

* Remove line 14.
* Replace line 13 with "for (int i : intList) {".
* Replace line 13 with "for (Iterator iter : intList) {".
* Replace the method declaration with "sum(List<int> intList)".
* Replace the method declaration with "sum(List<Integer> intList)"

**Q19**

What is the output for the below code?  
  
import java.util.Iterator;  
import java.util.TreeSet;  
public class Test {  
public static void main(String... args) {  
  
TreeSet s1 = new TreeSet();  
s1.add("one");  
s1.add("two");  
s1.add("three");  
s1.add("one");  
 Iterator it = s1.iterator();  
 while (it.hasNext() ) {  
 System.out.print( it.next() + " " );  
 }  
}  
}

*Single Choice - Select one correct answer from the options list.*

* one three two
* Runtime Exception
* one three two one
* one two three

**Q20**

Given the code fragment:  
  
public static vodi main(String args[]){  
ArrayList<String> list= new ArrayList<>();  
list.add("SE");  
list.add("EE");  
list.add("ME");  
list.add("SE");  
list.add("EE");  
  
list.remove("SE");  
System.out.println("Values are : "+ list);  
}

*Single Choice - Select one correct answer from the options list.*

* [ SE,EE,ME,EE]
* [ EE ME SE EE ]
* [ SE SE EE, EE]
* [EE ME EE]