Test 2

1. cannot be change after initialization

2. overridden

3. variable declaration

4. be instantiated

5. cannot be changed after initialization

6. are accessible in subclass

7. accessible only in package

8. A - (option a)

9. none of the above

10. option D

11. System

12. option b

13. option D

14. compile time error

15. option c

16. option c

17. java.util.hashset

18.

|  |  |
| --- | --- |
| Arraylist | Stack |
| Arraylist is faster performance | Compare to arraylist it is slower |
| Arraylist is non Synchronized method | Stack is Synchronized |
| No thread saftey | Provides thread safety |
|  |  |
|  |  |

19.

|  |  |
| --- | --- |
| **iterator** | **List Iterator** |
| We can traverse in only forward direction using Iterator. | we can traverse a List in both the directions |
| Iterator is used for traversing List and Set both. | ListIterator to traverse List only, we cannot traverse Set using ListIterator. |
| hasNext()  next()  remove() | add(E e)  hasNext()  hasPrevious()  next()  nextIndex()  set(E e)  etc |
|  |  |

20. **package** examofjava;

**import** java.util.HashSet;

**import** java.util.Iterator;

**public** **class** Javatet2 {

**public** **static** **void** main(String[] args) {

HashSet <String> jt = **new** HashSet <String>();

//insertion

jt .add("ShaShi");

jt.add("Sandeep");

jt.add("Rakesh");

jt.add("tharun");

jt.add("ketan");

System.***out***.println("insertion values "+ jt);

jt.remove("Sandeep");

jt.remove("ketan");

System.***out***.println("After deletion"+ jt);

System.***out***.println("Retrieving values from HashSet using Iterator");

*retrieveValuesFrom*(jt);

}

**public** **static** **void** retrieveValuesFrom(HashSet<String> jt) {

Iterator itr = jt.iterator();

**while**(itr.hasNext())

{

System.***out***.println(itr.next());

}

}

}

**Output**

insertion values [ShaShi, Sandeep, ketan, tharun, Rakesh]

After deletion[ShaShi, tharun, Rakesh]

Retrieving values from HashSet using Iterator

ShaShi

tharun

Rakesh