1. What will be the output of following code segment considering User class as below:

```
public class User {
    private int userId;
    private String userName;
//considering required constructor, getters & setters
}
```

```
public static void main(String[] args) {
    List<User> users=new ArrayList<>();
    users.add(new User(101));
    users.add(new User(101));
    users.add(new User(102));
    System.out.println(users.size());
    System.out.println(users.remove(new User(101)));
    System.out.println(users.size());
}
```

2. Consider following classes:

```
class Language
{
    private String languageName;
    private String description;
}

class Employee
{
    private Map<Language,Integer> skills; // stores languages known along
with experience in years
    private int empId;
    private String empName;
}
```

Assuming all the required getter,setters available in the above classes, Write a method that takes list of Employees and an employee id. It prints language name known by that employee in which he/she has experience more than 2 years. Method needs to be written in separate class.

3. **Consider the following Box class. Add few objects of Box to treeset in descending order of their volume.**

```
class Box{
    double length;
    double width;
    double height;
//assuming constructor, getters, setters available
}
public class Tester {
    public static void main(String[] args) {
//add code here
}
}
```

4. How can we sort elements of hashset? Give code snippet.
5. What will be the output of following program:

```
public class TestConcept {
    public static void main(String[] args) {
        Map<Project,String> managers=new HashMap<>();
        managers.put(new Project(10001,"Finnone"),"Hridesh");
        managers.put(new Project(10002,"Integration"),"Ritesh");
        Project myproject=new Project(10001,"Finnone");
        String myteamLead=managers.get(myproject);
        System.out.println(myteamLead);
    }
}
```

Considering following Project class:

```
class Project{
    private int projectId;
    private String projectName;
    Project(int projectId,String projectName)
    {
        this.projectName=projectName;
```

```
        this.projectId=projectId;
    }
    @Override
    public boolean equals(Object obj)
    {
        System.out.println("equals calling....");
        Project project=(Project) obj;
        if(this.projectId==project.projectId &&
this.projectName.equalsIgnoreCase(project.projectName))
            return true;
        return false;
    }
```

6. Consider the below code:

```
       TreeSet<String> treeSet=new TreeSet<>();
       treeSet.add("abc");
       treeSet.add("ABC");
       treeSet.add("xyz");
       treeSet.add("abC");
       treeSet.forEach(System.out::println);
```

It results:

```
ABC
abC
abc
xyz
```

What modification you'll do in the code so that same string in every case should be considered as one. In above code only two strings should be displayed:

```
abc
xyz
```

7. Why do we override equals and hashcode methods together?
8. What is the difference between hashset and treeset?
9. Consider the below code:

```
public class Test{
    public static void main(String[] args) {
        Map<String,Integer> gradeMap=new HashMap<>();
        gradeMap.put("A",1);
        gradeMap.put("B",2);
        gradeMap.put("C",3);
        gradeMap.put("C",4);
        gradeMap.put(null,5);

        for (Map.Entry<String, Integer> grade: gradeMap.entrySet())
        {
           System.out.println("key: " + grade.getKey() + " value: " +
grade.getValue());
        }
    }
}
```

What is Entry in above code? What will be the output?

10. What will be the output of below code:

```
    class Repository{
        private List<Integer> integerList;
        public void addInteger(int element){
            getIntegerList().add(element);
        }
        public void setIntegerList(List<Integer> integerList) {
            this.integerList = integerList;
        }
        public List<Integer>getIntegerList(){
            return integerList;
        }
    }
    public class Main {
        public static void main(String[] args) {
            Repository repository=new Repository();
            List<Integer> integerList=new ArrayList<>();
```

```
            repository.addInteger(10);
            System.out.println(repository.getIntegerList());
        }
    }
```