Group A

1. Given the following Java program:

```
import java.util.*;
public class Main
{
public static void main(String[] args) {
List num = new ArrayList(Arrays.asList(23, 16, 14, 33, 19, 6, 1));
System.out.println("List is "+num);
}
}
```

(a) Give the index values of all the odd numbers assuming zero-based indexing
(b) How many elements would be looked at when the list is traversed (from start to finish) until the value 19 was found?

2. Which of the following lists are syntactically correct in Java?

Try them out in to see if you were correct.

(a) List num = new ArrayList(Arrays.asList(1, 2, 3, 'four'));
(b) List num = new ArrayList(Arrays.asList(1, 2, [3, 4]));

3. Perform a series of list operations on the following list:

List fruit = new ArrayList (Arrays.asList('apple', 'banana', 'pear', 'cherry'));

to produce this updated list:

['Grapefruit', 'banana', 'Date', 'cherry', 'Orange']

Group B

1. Write a program to find out whether a given integer is present in an array or not.
2. Calculate the average marks from an array containing marks of all students in physics using a for-each loop.
3. Write a Java program to reverse an array.
4. Write a Java program to find the maximum element in an array.
5. Write a Java program to find whether an array is sorted or not.

Group C

1. Write a Java program to append the specified element to the end of a hash set.
2. Write a Java program to compare two sets and retain elements which are same on both sets.
3. Write a Java program to count the number of key-value mappings in a hash table
4. Write a Java program to get a collection view of the values contained in this map

(Optional)

Group D

1. Building a Rock Paper Scissor game in java

Ask the user to enter in their move.
Make a list of valid moves.
Check if the user entered a valid move by looking at the list of valid moves. (If the move is in the list, it is valid move)
Randomly generate the opponent's move. (Randomly choose one move from the list of valid moves)
Display the result to user
Use a loop to continue asking the user for their move.
Check if the user wants to quit.