1) Write a java program to display three highest elements from given array?
Input:
 7 2 4 9 1 6 3
Output:
 9 7 6

sol:

```
public class Test {
    public static void main(String[] args) {
        int[] array = {7, 2, 4, 9, 1, 6, 3};
        int firstMax, secondMax, thirdMax;

        firstMax = secondMax = thirdMax = Integer.MIN_VALUE;

        for (int i = 0; i < array.length; i++) {
            if (array[i] > firstMax) {
                thirdMax = secondMax;
                secondMax = firstMax;
                firstMax = array[i];
            } else if (array[i] > secondMax) {
                thirdMax = secondMax;
                secondMax = array[i];
            } else if (array[i] > thirdMax) {
                thirdMax = array[i];
            }
        }

        System.out.println("First Maximum: " + firstMax);
        System.out.println("Second Maximum: " + secondMax);
        System.out.println("Third Maximum: " + thirdMax);
    }
}
```

2) Write a java program to delete first occurrence of a given element?
Input:
 arr = 6 4 2 3 9 2 7 2 1
 element = 2
Output:
 6 4 3 9 2 7 2 1

sol:

```
public class Test {
    public static void main(String[] args) {
        int[] arr = {6, 4, 2, 3, 9, 2, 7, 2, 1};
        int element = 2;

        int index = 0;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == element) {
                index = i;
```

```
                break;
            }
        }
        if (index != 0) {
            for (int i = index; i < arr.length - 1; i++) {
                arr[i] = arr[i + 1];
            }
            arr[arr.length - 1] = 0;
        }
        for (int i = 0; i < arr.length - 1; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

3)Write a java program to perform sum of diagonal elements?
Input:
arr={ {1,2,3}
 {4,5,6},
 {7,8,9} };

sol:

```
public class Test {
    public static void main(String[] args) {
        int[][] arr = {
            {1, 2, 3},
            {4, 5, 6},
            {7, 8, 9}
        };

        int firstSum = 0;
        int secondSum = 0;

        for (int i = 0; i < arr.length; i++) {
            firstSum += arr[i][i];
            secondSum += arr[i][arr.length - 1 - i];
        }

        System.out.println("Sum of diagonal elements: " + firstSum);
        System.out.println("Sum of diagonal elements: " + secondSum);
    }
}
```

4) Write a java program to create custom exceptions?

sol:

Creating custom exceptions in Java allows you to define your own exception types that are more meaningful for your specific use case. Here's how you can create and use custom exceptions in Java.

example:

```
package mypack;

class ManikantaException extends RuntimeException//Exception
{

        public ManikantaException(String string) {
                super(string);
        }
}

public class Exc6 {

        public static void main(String[] args) {
         int i=20;
         int j=0;
         try {
                 j=18/i;
                 if(j==0)
                         //throw new ArithmeticException(" I dont want to print
zero");
                         throw new ManikantaException(" I dont want to print
zero");

        // }catch(ArithmeticException e) {
         }catch(ManikantaException e) {
                 j=18/1;
                 System.out.println("that is default output "+e);

         }
         catch(Exception e) {
                 System.out.println("Something went wrong");
         }
         System.out.println(j);
         System.out.println("Bye");
        }

}
```

5) Write a java program to handle multiple exceptions using single catch block?

sol:

In Java, you can handle multiple exceptions in a single catch block by using a pipe (|) to separate the exception types. This allows you to catch and handle different types of exceptions in one block of code.

```
public class MultipleExceptionsExample {
```

```
    public static void main(String[] args) {
        try {
            int[] numbers = {1, 2, 3};
            int result = numbers[3] / 0;  // This line can throw two exceptions
            System.out.println("Result: " + result);
        } catch (ArithmeticException | ArrayIndexOutOfBoundsException e) {
            System.out.println("An exception occurred: " + e.getMessage());
        }
    }
}
```