Assignment-21

Arrays, Strings, Abstract Classes & File Handling

1. Write a Java program to calculate the average value of array elements.

```java
package Assignment21;

public class Avg {

    public static void main(String[] args) {

        int[] number = new int[]{10, 20, 30, 50, -15, -25,};

           int sum = 0;
           for(int i = 0; i < number.length; i++)
            sum = sum + number[i];

            double average = sum / number.length;
            System.out.println("Average value of array elements : " + average);
    }

}
```

```
Average value of array elements : 11.0
```

2. Write a Java program to test if an array contains a specific value.

```java
package Assignment21;

public class Array_value {

    public static boolean contains(int[] a, int item) {

        for (int n : a) {
             if (item == n)
             {
                return true;
             }
          }
          return false;
          }
      public static void main(String[] args) {
             int[] my_array1 =
             {11, 123, 3455, 898, 777, 2000, 5567, 990,};

          System.out.println(contains(my_array1, 777));
          System.out.println(contains(my_array1, 999));
      }
}
```

```
true
false
```

3. Write a Java program to remove a specific element from an array.

```java
package Assignment21;
import java.util.*;

public class Remove {

    public static void main(String[] args) {
        int[] my_array = {5, 10, 15, 20, 25, 50, 100};
        System.out.println("Original Array : "+Arrays.toString(my_array));

           int removeIndex = 0;

           for(int i = removeIndex; i < my_array.length -1; i++)
           {
                my_array[i] = my_array[i + 1];
           }
           System.out.println("After removing the first element: "+Arrays.toString(my_array));
    }
}
```

```
Console
<terminated> Remove [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-20.
Original Array : [5, 10, 15, 20, 25, 50, 100]
After removing the first element: [10, 15, 20, 25, 50, 100, 100]
```

4. Write a Java program to copy an array by iterating the array.

```java
package Assignment21;

public class Iterating {

    public static void main(String[] args) {
        int a[] = {5, 10, 15, 20, 25};
        int b[] = new int[a.length];

        b = a;

        System.out.println("Elements of array a[] ");
        for (int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");

        System.out.println("\nElements of array b[] ");
        for (int i = 0; i < b.length; i++)
            System.out.print(b[i] + " ");
    }
}
```

```
Console
<terminated> Iterating [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-20
Elements of array a[]
5 10 15 20 25
Elements of array b[]
5 10 15 20 25
```

5. Write a Java program to insert an element (specific position) into an array.

```java
package Assignment21;
import java.util.*;

public class Insert {

    public static void main(String[] args) {
    int[] my_array = {1, 2, 3, 4, 6, 7};
    int Index_position = 4;
    int newValue    = 5;
    System.out.println("Original Array : "+Arrays.toString(my_array));
    for(int i=my_array.length-1; i > Index_position; i--){
    my_array[i] = my_array[i-1];
    }
    my_array[Index_position] = newValue;
    System.out.println("New Array: "+Arrays.toString(my_array));
    }
}
```

```
<terminated> Insert [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 5:14:07 pm
Original Array : [1, 2, 3, 4, 6, 7]
New Array: [1, 2, 3, 4, 5, 6]
```

6. Write a Java program to find the duplicate values of an array of integer values.

```java
package Assignment21;

public class Duplicate {

    public static void main(String[] args) {
        int [] array = new int [] {10, 11, 12, 11, 10, 10, 50};
        System.out.println("Duplicate elements in given array: ");
        for(int i = 0; i < array.length; i++) {
            for(int n = i + 1; n < array.length; n++) {
                if(array[i] == array[n])
                    System.out.println(array[n]);
            }
        }
    }
}
```

```
<terminated> Duplicate [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 5:20:10
Duplicate elements in given array:
10
10
11
10
```

7. Write a Java program to find the common elements between two arrays of integers.

```java
package Assignment21;

public class common {

    public static void main(String[] args) {
        int[] array1 = {7, 8, 9, 33, 55};
          int[] array2 = {10, 11, 33, 9, 55, 100};

          for(int i = 0;i < array1.length; i++) {
             for(int j = 0; j < array2.length; j++) {
                if(array1[i] == array2[j]) {
                   System.out.println(array1[i]);
                }
             }
          }
    }
}
```

```
<terminated> common [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 5:24:57
9
33
55
```

8. Write a Java program to remove duplicate elements from an array.

```java
package Assignment21;

public class removeD {
        public static int removeDuplicateElements(int arr[], int n){
            if (n==0 || n==1){
                return n;
            }
            int[] temp = new int[n];
            int j = 0;
            for (int i=0; i<n-1; i++){
                if (arr[i] != arr[i+1]){
                    temp[j++] = arr[i];
                }
             }
            temp[j++] = arr[n-1];
            for (int i=0; i<j; i++){
                arr[i] = temp[i];
            }
            return j;
        }

        public static void main (String[] args) {
            int arr[] = {1, 2, 2, 3, 3, 3, 4, 4, 4, 4};
            int length = arr.length;
            length = removeDuplicateElements(arr, length);
            for (int i=0; i<length; i++)
                System.out.print(arr[i]+" ");
        }
}
```

```
<terminated> removeD [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 5:29:39
1 2 3 4
```

9. Write a Java program to find the second largest element in an array.

```java
package Assignment21;

public class SecondLargest {

    public static int getSecondLargest(int[] a, int total){
        int temp;
        for (int i = 0; i < total; i++)
                {
                    for (int j = i + 1; j < total; j++)
                    {
                        if (a[i] > a[j])
                        {
                            temp = a[i];
                            a[i] = a[j];
                            a[j] = temp;
                        }
                    }
                }
               return a[total-2];
        }
        public static void main(String args[]){
        int a[]={11, 22, 33, 99, 9};
        System.out.println("Second Largest: "+getSecondLargest(a,4));
        }
}
```

```
Console
<terminated> SecondLargest [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 5:
Second Largest: 33
```

10. Write a Java program to find smallest and second smallest elements of a given array.

```java
package Assignment21;

public class smallest {

    public static void main(String[] args) {
        int arr[] = {-1, 1, 0, 1, 2,};

          int first_element, second_element, arr_size = arr.length;
          if (arr_size < 2)
            {
                System.out.println("Array size less than two.");
                return;
            }
          first_element = second_element = Integer.MAX_VALUE;
          for (int i = 0; i < arr_size ; i ++)
            {
              if (arr[i] < first_element)
              {
                  second_element = first_element;
                  first_element = arr[i];
              }
              else if (arr[i] < second_element && arr[i] != first_element)
                  second_element = arr[i];
              }
          if (second_element == Integer.MAX_VALUE)
              System.out.println("No second smallest element.");
            else
                System.out.println("The smallest element is " + first_element + " and second Smallest  element is " + second_eleme
          }
}
```

```
Console
<terminated> smallest [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 5:42:11 pm – 5:42:12 pm) [pid: 10512]
The smallest element is -1 and second Smallest  element is 0.
```

11. Write a Java program to get the character (Unicode code point) at the given index within the String.

```java
package Assignment21;

public class Char {

    public static void main(String[] args) {
        String str = "Rana Akshat";
        System.out.println("Original String : " + str);
        int val1 = str.codePointAt(1);
        int val2 = str.codePointAt(2);

        System.out.println("Character(unicode point) = " + val1);

        System.out.println("Character(unicode point) = " + val2);
        }
}
```

```
Original String : Rana Akshat
Character(unicode point) = 97
Character(unicode point) = 110
```

12. Write a program to remove first occurrence of a character from string.

```java
package Assignment21;

public class occurance {

    public static void main(String[] args) {
        String st = "Height of a person!";
           System.out.println("String: "+st);
           String str = st.replaceFirst("(?:H)+", "L");
           System.out.println("String after replacing a characters first occurrence: "+str);
    }
}
```

```
String: Height of a person!
String after replacing a characters first occurrence: Leight of a person!
```

13. Write a program to remove last occurrence of a character from string.

```java
package Assignment21;

public class lastOcc {

    public static void main(String[] args) {
        String string = "Java is a programming language !";
        StringBuffer sb= new StringBuffer(string);
        sb.deleteCharAt(sb.length()-1);
        System.out.println(sb);
        }
}
```

```
Java is a programming language
```

14. Write a program to remove all occurrences of a character from string.

```java
package Assignment21;

public class Removeall {

    public static void removeChar(String word, char ch) {
        word = word.replace(Character.toString(ch), "");
        System.out.println(word);
    }

    public static void main(String[] args) {
        String word = "java is a base";
        removeChar(word, 'a');
        }
}
```

```
<terminated> Removeall [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 6:01:14
jv is  bse
```

15. Write a Java program to replace a specified character with another character.

```java
package Assignment21;

public class Replace {

    public static void main(String[] args) {
        String s1 = "My name is Rana Akshat";
          System.out.println("String: "+s1);
          String s2 = s1.replace(' ', '_');
          System.out.println("Updated string: "+s2);
          }
}
```

```
<terminated> Replace [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (22-Mar-2023, 6:07:12 p
String: My name is Rana Akshat
Updated string: My_name_is_Rana_Akshat
```

16. Write a program to trim leading white space characters in a string.

SecondLarge... smallest.java Char.java occurance.java lastOcc.java Removeall.java Replace.java

```java
package Assignment21;

public class trim {

    public static void main(String[] args) {
        String s1 = " Hello ";
        System.out.println(s1);
        System.out.println(s1.trim());
        }
}
```

Console

```
<terminated> trim [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-093
 Hello
Hello
```

17. Write a program to trim trailing white space characters in a string.

smallest.java Char.java occurance.java lastOcc.java Removeall.java Replace.java trim.java *trimTraili...

```java
package Assignment21;

public class trimTrailing {

    public static void main(String[] args) {
        String s1 = "  Hello  ";
        System.out.println(s1);
        System.out.println(s1.trim());

        String s2 = "    I  am  here    ";
        System.out.println(s2);
        System.out.println(s2.trim());

    }
}
```

Console

```
<terminated> trimTrailing [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.e
  Hello
Hello
    I  am  here
I  am  here
```

18. Write a program to print the word & count of words present in a given

sentence.

```java
package Assignment21;
import java.util.*;
public class Count {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Input the string: ");
        String st = sc.nextLine();

        System.out.print("Number of words in the string: " + count_Words(st)+"\n");
        }
    public static int count_Words(String str)
        {
        int count = 0;
        if (!(" ".equals(str.substring(0, 1))) || !(" ".equals(str.substring(str.length() - 1))))
        {
            for (int i = 0; i < str.length(); i++)
            {
                if (str.charAt(i) == ' ')
                {
                    count++;
                }
            }
            count = count + 1;
        }
        return count;
        }
}
```

```
Input the string: My name is Rana
Number of words in the string: 4
```

19. Check whether a string entered by user is a palindrome string or not.

```java
package Assignment21;

public class Palindrome {

    public static void main(String[] args) {
        String s = "mom";
        String rev = "";
        for (int i = s.length()-1; i >=0 ; i--)
            rev=rev+s.charAt(i);
        if(s.equals(rev))
            System.out.println("String is palindrome");
        else
            System.out.println("String is not palindrome");
        }
}
```

```
String is palindrome
```

20. What will happen if we do not override all abstract methods in subclass ?

Ans: A compile time error will be generated for each abstract method (that you don't override) saying “subclass_name is not abstract and does not override abstract method abstractmethod_name in classname”.

21. What is the difference between Abstraction and Encapsulation ?

Abstraction :

Abstraction is hiding the details and implementation of the code.

Abstraction is a design level process.

Abstraction is concerned about what a class instance can do, instead of the implementation of the class.

Encapsulation :

Encapsulation is hiding the data and controlling the visibility of the code.

Encapsulation is an implementation level process.

Encapsulation helps in data binding and control over maintaining the transparency of the data.

22. Why abstract class has constructor even though you cannot create object ?

Ans: Abstract class has constructor even though we cannot create object because it's abstract and an object is concrete. An abstract class is sort of like a template, or an empty/partially empty structure, you have to extend it and build on it before you can use it. abstract class has a protected constructor (by default) allowing derived types to initialize it.

23. What is Out Of Memory Error in Exception Handling ?

Ans: Out Of Memory Error usually means that you're doing something wrong, either holding onto objects too long or trying to process too much data at a time. Sometimes, it indicates a problem that's out of your control, such as a third-party library that caches strings or an application server that doesn't clean up after deploys. And sometimes, it has nothing to do with objects on the heap.

24. What is the difference between throws and throw in java ?

Throws :

The throws keyword is used in the function signature. It is used when the function has some statements that can lead to exceptions.

The throws keyword can be used to declare multiple exceptions, separated by a comma. Whichever exception occurs, if matched with the declared ones, is thrown automatically then.

Syntax of throws keyword includes the class names of the Exceptions to be thrown. Syntax wise throws keyword is followed by exception class names.

throws keyword is used to propagate the checked Exceptions only.

Throw :

The throw keyword is used inside a function. It is used when it is required to throw an Exception logically.

The throw keyword is used to throw an exception explicitly. It can throw only one exception at a time.

Syntax of throw keyword includes the instance of the Exception to be thrown. Syntax wise throw keyword is followed by the instance variable.

throw keyword cannot propagate checked exceptions. It is only used to propagate the unchecked Exceptions that are not checked using the throws keyword.

25. Give some examples of Checked exceptions ?

Ans: Checked exceptions are the subclass of the Exception class. These types of exceptions need to be handled during the compile time of the program. These exceptions can be handled by the try-catch block or by using throws keyword otherwise the program will give a compilation error.

Examples are - IOException, SQLException, ClassNotFoundException, etc

26. Give some examples of Unchecked exceptions ?

Ans: Unchecked exceptions are not checked at compile time. In Java, exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked.

Examples are - ArithmeticException, ClassCastException, NullPointerException, IllegalArgumentException, etc.

27. List the Methods in the Throwable class ?

Ans: Java Throwable class provides several methods like addSuppressed(), fillInStackTrace(), getMessage(), getStackTrace(), getSuppressed(), toString(), printStackTrace() etc.

28. What is Array Index Out Of Bounds Exception in java ?

Ans: The ArrayIndexOutOfBoundsException is a Runtime Exception thrown

only at runtime. The Java Compiler does not check for this error during the compilation of a program.

29. Will the code compile successfully? If yes, what will be the output ?

```
public abstract class A {
abstract void m1();
}
public class B extends A {
 void m1(){
 System.out.println("m1 in class B");
 }
}
public class Test {
public static void main(String[] args) {
B b = new B();
 b.m1();
 }
}
```

Ans: Yes, Code will be compiled successfully

output :

m1 in class B

31.Consider the below given code.

```
public abstract class A {

abstract void m1();

void m2(){

 System.out.println("One”);

 }

}
```

How to call m2() method in the above code ?

Ans: We will create a class B and we will extend the class A into Class B .Then we will create an object of class B and using "objectname.m2()", we can access the m2() function.