**Java Programming Lab**

**(ETCS-357)**

****

**Name:** Guneet Malhotra

**Branch:** Information Technology (IT)

**Shift:** 2nd (Evening)

**Semester:** 5th

**Submitted To:** Mr. Sitender Malik

**Ques1. WAP to print “Hello World” in JAVA.**

Java Code:

public class HelloWorld {

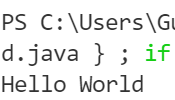
    public static void main(String[] args) {

        System.out.println("Hello World");

    }

}

**Output:**



**Ques2. WAP a program to print whether the number is even or odd in JAVA.**

Java Code:

import java.util.Scanner;

public class EvenOdd{

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        int n =scn.nextInt();

        if(n==0) {

            System.out.println("The number is neither even nor odd");

        }

        else if(n%2 == 0){

            System.out.println("Even");

        }

        else{

            System.out.println("Odd");

        }

        scn.close();

    }

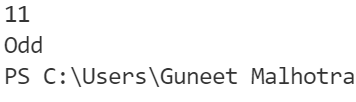
}

**Output:**

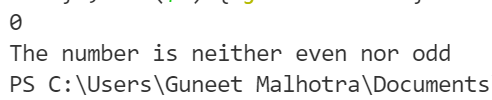
For even numbers:



For odd numbers:



For Zero:



**Ques3. WAP to find the factorial of a number in JAVA.**

Java Code:

import java.util.\*;

public class Factorial {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();

        System.out.println("The factorial of "+ n + " is :" + factorial(n));

        scn.close();

    }

    public static int factorial(int n){

        int fact=1;

        for(int i=2;i<=n;i++) {

            fact \*= i;

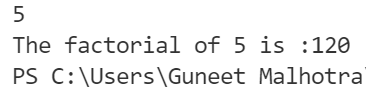
        }

        return fact;

    }

}

**Output:**



**Ques4. WAP to check whether a number is prime or not in JAVA.**

Java Code:

import java.util.\*;

public class PrimeNumber {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();

        scn.close();

        if(isPrime(n)) {

            System.out.println(n + " is a prime number");

        } else {

            System.out.println(n + " is not a prime number");

        }

    }

    public static boolean isPrime(int n){

        for(int i=2; i\*i<=n;i++)  {

            if(n%i == 0) {

                return false;

            }

        }

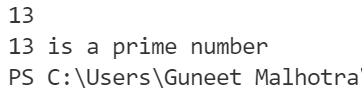
        return true;

    }

}

**Output:**

For Prime Numbers:



For Composite Numbers



**Ques5. WAP in JAVA to check whether a year is a leap year or not.**

Java Code

import java.util.\*;

public class LeapYear {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        int year = scn.nextInt();

        scn.close();

        if(isLeap(year)) {

            System.out.println(year + " is a leap year");

        } else {

            System.out.println(year + " is not a leap year");

        }

    }

    public static boolean isLeap(int year) {

        if((year % 4) == 0) {

            if((year % 100) == 0) {

                if((year % 400) == 0) {

                    return true;

                }

                else {

                    return false;

                }

            }

            else {

                return true;

            }

        }

        return false;

    }

}

**Output:**

For a leap year:



For a non-leap year



**Ques6. WAP in JAVA to print the Fibonacci series up to N terms (Take N input from the user)**

Java Code

import java.util.\*;

public class fibonnaciSeries {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        System.out.println("Enter the number of terms that you want for fibonnaci series");

        int n = scn.nextInt();

        scn.close();

        solution(n);

    }

    public static void solution(int n) {

        int a=0;

        int b=1;

        if(n==0) {

            return;

        }

        else if(n==1) {

            System.out.println("0");

            return;

        }

        else if(n==2) {

            System.out.println("0");

            System.out.println("1");

            return;

        }

        System.out.println(a);

        System.out.println(b);

        for(int i=3;i<=n;i++) {

            int c = a + b;

            System.out.println(c);

            a = b;

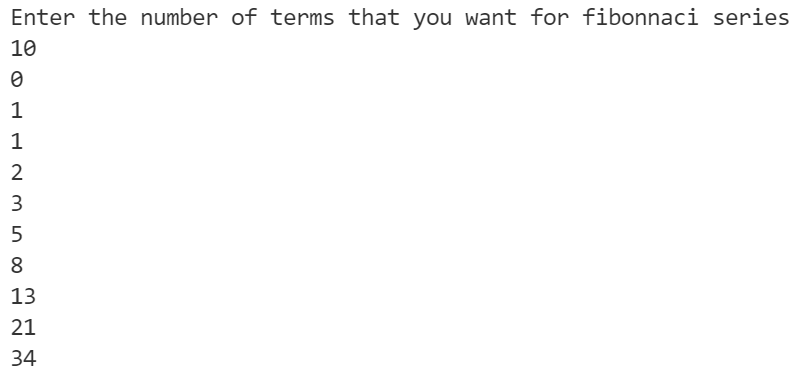
            b = c;

        }

    }

}

**Output:**

****

**Ques7. WAP to find the factorial of a number using recursion**

Java Code

import java.util.\*;

public class factorialUsingRecursion {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();

        scn.close();

        System.out.println("The factorial of " + n + " is : " + factorial(n));

    }

    public static int factorial(int n) {

        if(n == 0) {

            return 1;

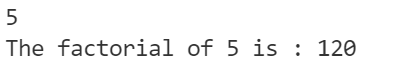
        }

        return n\*factorial(n-1);

    }

}

**Output:**

****

**Ques8. WAP to find a number to the power other in JAVA**

Java Code:

import java.util.\*;

public class Power {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        int n = scn.nextInt();

        int p = scn.nextInt();

        scn.close();

        System.out.println(power(n,p));

    }

    public static int power(int n, int p) {

        if(p == 0) {

            return 1;

        }

        if(p % 2 == 0) {

            return power(n,p/2) \* power(n,p/2);

        }

        return n\* power(n,p/2) \* power(n,p/2);

    }

}

**Output:**

****

**Ques10. WAP in JAVA to show implicit and explicit type casting.**

Java Code:

public class DataTypes {

    public static void main(String[] args) {

        //implicit typecasting

        int x = 10;

        long y = x;  //automatic/implicit type casting

        System.out.println("Integer: " + x);

        System.out.println("Long Type Casted" + y);

        //explicit typecasting

        long l = 10000;

        int i = (int)l;

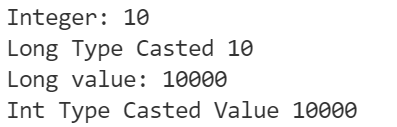
        System.out.println("Long value: " + l);

        System.out.println("Int Type Casted Value" + i);

    }

}

**Output:**



**Ques11. WAP in JAVA to take input 10 numbers as command line arguments and sort them using any sorting algorithm.**

Java Code

import java.util.Arrays;

public class CommandLine {

    public static void main(String[] args) {

         int[] arr = new int[args.length];

         for(int i=0;i<arr.length;i++) {

             arr[i] = Integer.parseInt(args[i]);

         }

         Arrays.sort(arr);

         for(int i=0;i<arr.length;i++) {

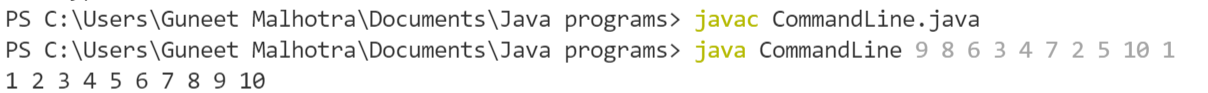
            System.out.print(arr[i] + " ");

         }

    }

}

**Output:**

****

**Ques12. WAP in JAVA to calculate the sum of first 100 natural numbers using recursion.**

Java code

public class SumFrom1To100 {

    public static void main(String[] args) {

        System.out.println(sum(100));

    }

    public static int sum (int i) {

        if(i==1){

            return 1;

        }

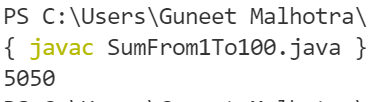
        int tsum = i + sum(i-1);

        return tsum;

    }

}

**Output:**

****

**Ques13 WAP in JAVA to illustrate the use of classes and objects.**

Java Code

public class ClassIntro {

    static class Student {

        int marks;

        int age;

        String name;

        Student() {

        }

        Student(int marks,int age,String name) {

            this.marks = marks;

            this.age = age;

            this.name = name;

        }

        void printData() {

            System.out.println("The name of the student is: " + name + " and the age is: " + age +" and the marks are: " + marks);

        }

    }

    public static void main(String[] args) {

        Student s = new Student();

        s.printData();

        Student s2 = new Student(90, 17, "George");

        s2.printData();

        Car c1 = new Car();

        c1.printData();

        Car c2 = new Car("Toyota", "Black");

        c2.printData();

    }

}

class Car {

    String cName;

    String color;

    Car() {

    }

    Car(String cName,String color) {

        this.cName=cName;

        this.color=color;

    }

    void printData()

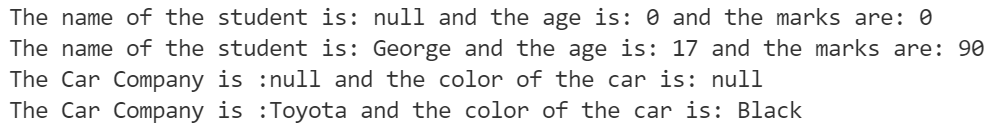
    {

        System.out.println("The Car Company is :" + cName + " and the color of the car is: " + color);

    }

}

**Output:**

****

**Ques 14. WAP in JAVA to show different protection modes with the help of packages.**

We have created two packages namely “pack” and “myPack”. We have created the first class Main1 in the package “pack” and the second class Main2 in the package “myPack”. Also, we have imported the package pack to the program Main2.java as shown below:

Source Code for Main1.java

package pack;

public class Main1 {

    protected void main1Message() {

        System.out.println("protected function called");

    }

    public static void main(String[] args) {

        A obj = new A();

        obj.a=10; // cannot be accessed as a is private

    }

}

class A {

    private int a=0;

    void msg() {

        System.out.println("Hello");

    }

}

Source Code for Main2.java

package my*P*ack;

import pack.\*;

public class Main2 {

    public static void main(String[] args) {

        A obj = new A(); //cannot be accessed as class A is in differnt package "pack" and is default

        Main1 m1 = new Main1(); //object can be made as class Main1 is public

        m1.main1Message(); //cannot be accesses as it is protected

    } }

**Ques15. WAP in Java to show multilevel inheritance.**

Source Code:

public class MultiLevelInheritance {

    public static void main(String[] args) {

        System.out.println("........................Class A objects ................");

        A aObj = new A(); // class A defualt constructor called

        A obj1 = new A(10);

        obj1.print();

        System.out.println("........................Class B Objects .................");

        B bObj = new B(); // class B default constructor called

        B obj2 = new B(10, 20);

        obj2.print();

        System.out.println("........................Class C Objects .................");

        B cObj = new C(); // class C default constructor called

        C obj3 = new C(10, 20, 30);

        obj3.print();

    }

}

class A {

    int a = 0;

    A() {

        System.out.println("A class constructor called");

    }

    A(int a) {

        this.a = a;

    }

    void print() {

        System.out.println("The value of a is = " + a);

    }

}

class B extends A {

    int b = 0;

    B() {

        System.out.println("Class B constructor called");

    }

    B(int a, int b) {

        super(a);

        this.b = b;

    }

    void print() {

        super.print();

        System.out.println("The value of b is = " + b);

    }

}

class C extends B {

    int c = 0;

    C() {

        System.out.println("Class C constructor called");

    }

    C(int a, int b, int c) {

        super(a, b);

        this.c = c;

    }

    void print() {

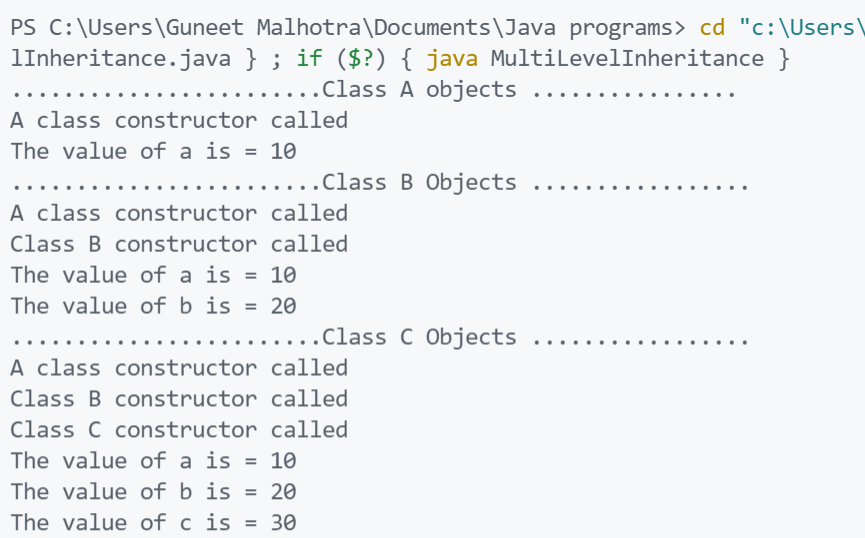
        super.print();

        System.out.println("The value of c is = " + c);

    }

}

**Output:**

****

**Ques16. WAP in Java to show multiple inheritance.**

Java Code:

package OOPS;

public class InterfaceExample {

    public static void main(String[] args) {

        SmartPhone sp1 = new SmartPhone();

        Phone p1 = new SmartPhone();

        ICamera c1 = new SmartPhone();

        IMusicplayer m1 = new SmartPhone();

        sp1.videocall();

        p1.call();

        p1.message();

        c1.click();

        c1.record();

        m1.play();

        m1.pause();

        m1.stop();

    }

}

class Phone {

    void call() {

        System.out.println("call");

    }

    void message() {

        System.out.println("Message");

    }

}

interface ICamera {

    void click();

    void record();

}

interface IMusicplayer {

    void play();

    void pause();

    void stop();

}

class SmartPhone extends Phone implements ICamera, IMusicplayer {

    void videocall() {

        System.out.println("Video call");

    }

    @Override

    public void click() {

        System.out.println("Picture click");

    }

    @Override

    public void record() {

        System.out.println("Record video");

    }

    @Override

    public void play() {

        System.out.println("Play music");

    }

    @Override

    public void pause() {

        System.out.println("Pause Music");

    }

    @Override

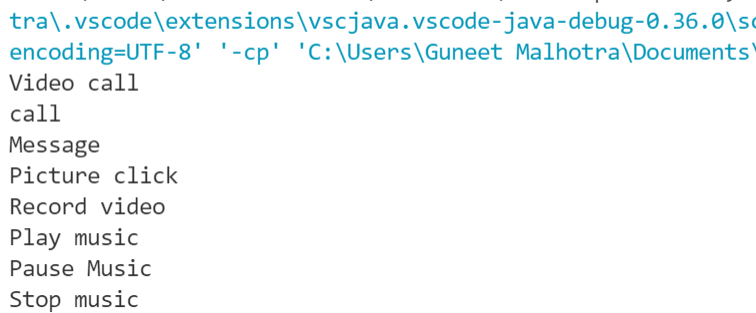
    public void stop() {

        System.out.println("Stop music");

    }

}

**Output:**

****

**Ques17. WAP in Java to show the nesting of classes (Inner Classes).**

Java Code

public class NestingOfClasses {

    public static void main(String[] args) {

        Outer obj1 = new Outer(10, 20);

        obj1.showData();

        Outer.Inner1 obj2 = obj1.new Inner1(30);

        obj2.showData();

        Outer.Inner2 obj3 = new Outer.Inner2(40);

        obj3.showData();

        Outer.Inner3.z = 100;

        System.out.println(Outer.Inner3.z);

    }

}

class Outer {

    private int x, y;

    Outer() {

        System.out.println("Outer class default constructor called");

    }

    Outer(int x, int y) {

        this.x = x;

        this.y = y;

    }

    void showData() {

        System.out.println("the value of x is:" + x + " and the value of y is: " + y);

    }

    class Inner1 {

        int z = 0;

        Inner1() {

            System.out.println("Inner class default constructor called");

        }

        Inner1(int z) {

            this.z = z;

        }

        void showData() {

            System.out.println("The value of x is: " + x + " the value of y is: " + y + " and z is: " + z);

        }

    }

    static class Inner2 {

        int z = 0;

        Inner2() {

            System.out.println("Inner class default constructor called");

        }

        Inner2(int z) {

            this.z = z;

        }

        void showData() {

            System.out.println("The value of z is: " + z);

        }

    }

    static class Inner3 {

        static int z = 0;

        Inner3() {

            System.out.println("Inner class default constructor called");

        }

        Inner3(int a) {

            z = a;

        }

        void showData() {

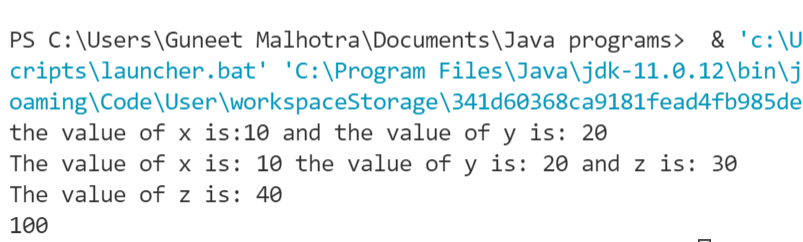
            System.out.println("The value of z is: " + z);

        }

    }

}

**Output:**

****

**Ques.18 WAP in Java to show the use of try, catch, finally, throw and throws keywords.**

Java Code:

package Exception\_Handling;

public class AllKeyWords {

    public static void main(String[] args) {

        meth1(); // error will be handled and message will be displayed

        int[] arr = { 1, 2, 3, 4, 5 };

        try {

            meth2(arr); // error will be caught here

        } catch (Exception e) {

            System.out.println(e);

        }

        meth(); // error is not handled so the program will crash but display the final message

                // first

    }

    static void meth() {

        try {

            System.out.println(10 / 0);

        } finally {

            System.out.println("Final message from meth");

        }

    }

    static void meth1() {

        try {

            System.out.println(10 / 0);

        } catch (ArithmeticException e) {

            System.out.println(e);

        }

    }

    static void meth2(int[] arr) throws ArrayIndexOutOfBoundsException {

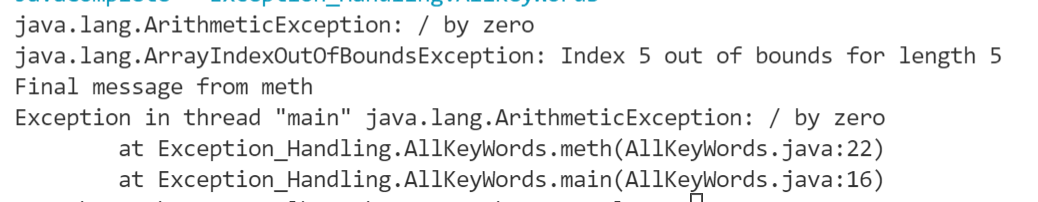
        System.out.println(arr[0] / arr[arr.length]);

        throw new ArrayIndexOutOfBoundsException();

    }

}

**Output:**

****

**Ques19. WAP in Java to implement a user defined exception.**

We are going to write an exception which generates when a user comes to our bank to open an account and he/she has less than 5000 rupees with them as the account should have a minimum balance of Rs 5000. This exception will be called LowBalanceException.

Java Code

package Exception\_Handling;

public class UserDefinedException {

    public static void main(String[] args) {

      Account a1 = new Account(500);

      Account a2 = new Account();

      a2.setBalance(500);

      Account a3 = new Account(10000);

      System.out.println("a1 balance = " + a1.getBalance() + "  a2 balance = " + a2.getBalance() + " a3 balance = " + a3.getBalance());

    }

}

class Account {

    private int balance;

    Account() {

        balance = 5000;

    }

    Account(int balance) {

        try {

            if(balance>=5000) {

                this.balance = balance;

                System.out.println("The account is created and the balance is set to: "+ balance);

            } else {

                this.balance=0;

                System.out.println("Account can not be created");

                throw new LowBalanceException();

            }

        } catch(LowBalanceException e) {

            System.out.println(e);

        }

    }

    void setBalance(int balance) {

        try {

            if(balance>=5000) {

                this.balance = balance;

                System.out.println("The account is created and the balance is set to: "+ balance);

            } else {

                this.balance=0;

                System.out.println("Account can not be created");

                throw new LowBalanceException();

            }

        } catch(LowBalanceException e) {

            System.out.println(e);

        }

    }

    int getBalance() {

        return balance;

    }

}

class LowBalanceException extends Exception {

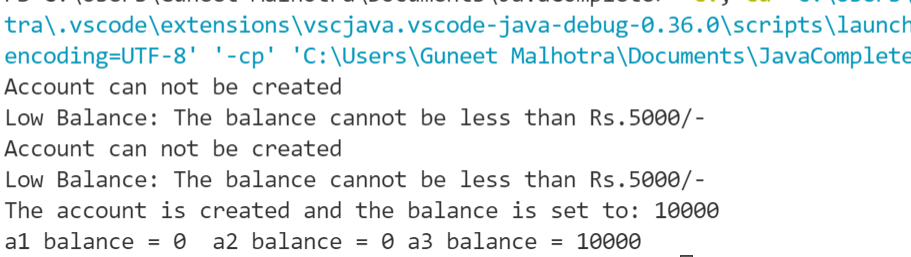
    public String toString() {

        return "Low Balance: The balance cannot be less tahn Rs.5000/-";

    }

}

**Output:**



**Ques20. WAP in Java to show a single thread.**

Java Code:

package Java\_Multithreading;

public class SingleThread {

    public static void main(String[] args) {

        Thread t = Thread.currentThread();

        t.setName("My Main Thread");

        t.setPriority(7);

        System.out.println(t);

        System.out.println(t.getName());

        System.out.println(t.getPriority());

    }

}

Output:

Text

Description automatically generated with medium confidence

**Ques21. WAP a program in Java to show multithreading and isAlive() and join() operations.**

Java Code:

package Java\_Multithreading;

class DemoThread extends Thread {

    public DemoThread(String name) {

        super(name);

        setPriority(MAX\_PRIORITY);

    }

}

class DemoThread2 extends Thread {

    public void run() {

        int count = 1;

        while (true) {

            System.out.println(count);

            count++;

            try {

                Thread.sleep(100);

            } catch (InterruptedException e) {

                System.out.println(e);

            }

        }

    }

}

public class ThreadClassFunctions {

    public static void main(String[] args) {

        DemoThread t = new DemoThread("Thread 1");

        System.out.println("ID " + t.getId());

        System.out.println("NAME " + t.getName());

        System.out.println("PRIORITY " + t.getPriority());

        t.start();

        System.out.println("STATE " + t.getState());

        System.out.println("ALIVE " + t.isAlive());

        DemoThread2 t2 = new DemoThread2();

        try {

            Thread.sleep(100);

        } catch (Exception e) {

        }

        t2.setDaemon(true);

        t2.start();

        // t2.interrupt();

        Thread mainThread = Thread.currentThread();

        try {

            mainThread.join(); // Now main will not terminate till the daemon thread is terminated

        } catch (Exception e) {

        }

    }

}

Output:

Text, letter

Description automatically generated

**Ques22 WAP in Java to use String and String Buffer classes.**

Java Code:

public class StringAndStringBuffer {

    public static void main(String[] args) {

        String str = "Hello"; // immutable

        System.out.println(str.length());

        StringBuffer sb = new StringBuffer(str); // mutable

        System.out.println(sb.toString()); // also stringBuffers are not synchronised

    }

}

**Output:**

**Text

Description automatically generated with low confidence**

**Ques23. WAP in java to take ten names as input from the console and display them on the screen using Stream classes in java.**

Java Code

package InputOutputStreams;

import java.io.\*;

public class ReadNames {

    public static void main(String[] args) throws IOException {

        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

        String[] parts = new String[10];

        for (int i = 0; i < 10; i++) {

            parts[i] = reader.readLine();

        }

        for (int i = 0; i < 10; i++) {

            System.out.print(parts[i] + " ");

        }

    }

}

Output:

Text

Description automatically generated with medium confidence

**Ques23. WAP in Java to read the data from one file and copy to another file using Byte Stream classes as well as Character Stream Classes.**