

```

1 package sum;
2 public class Sum extends Thread {
3     int from;
4     int where;
5     double result = 0;
6     public Sum(int from, int where) {
7         this.from = from;
8         this.where = where;
9     }
10    public void run() {
11        for (int i = from; i <= where; i++) {
12            result += Math.log(a: Math.sqrt(a: i));
13        }
14    }
15    public double getResult() {
16        return result;
17    }
18    public static void main(String[] args) {
19        Sum t1 = new Sum(from: 0, where: 250000000);
20        Sum t2 = new Sum(from: 250000001, where: 500000000);
21        Sum t3 = new Sum(from: 500000001, where: 750000000);
22        Sum t4 = new Sum(from: 750000001, where: 1000000000);
23        try {
24            t1.start(); t1.join();
25            t2.start(); t2.join();
26            t3.start(); t3.join();
27            t4.start(); t4.join();
28            double r = t1.getResult() + t2.getResult();
29            System.out.println("Result = " + r);
30        } catch (Exception e) {}
31    }
32 }

```

4. จงเขียนโปรแกรม Sum.java โดยโปรแกรมจะรับอาร์กิวเมนต์ที่ตัวก็ได้ที่เป็นจำนวนเต็ม ซึ่งโปรแกรมจะคำนวณผลรวมของทุกอาร์กิวเมนต์และเขียนผลลัพธ์จากการคำนวณลงไปไฟล์ชื่อ output.txt ถ้าอาร์กิวเมนต์บางตัวไม่ใช่จำนวนเต็มจะให้โปรแกรมแสดงคำว่า "Error args" บนหน้าจอ และหยุดการทำงานทันที (12 คะแนน)

ตัวอย่างการใช้งานโปรแกรม:

คำสั่ง	ผลลัพธ์
java Sum 5 4 9 1	จะได้ไฟล์ output.txt ภายในจะมีเลข 19 (เกิดจาก 5+4+9+1)
java Sum	จะได้ไฟล์ output.txt ภายในจะมีเลข 0
java Sum 1 A 2	แสดงคำว่า Error args บนหน้าจอ ไม่เกิดไฟล์ output.txt

```

1 package badthread;
2 public class BadThread {
3     public static void main(String[] args) {
4         BankAccount bankAcct = new BankAccount(money: 1000);
5         BankBranch b1 = new BankBranch(bankAcct, method:"deposit", money: 10000);
6         BankBranch b2 = new BankBranch(bankAcct, method:"withdraw", money: 10000);
7         b1.start();
8         b2.start();
9         try {
10             b1.join();
11             b2.join();
12         } catch (Exception e) {}
13         System.out.println("Balance = "+bankAcct.getBalance());
14     }
15 }
16 }

```

```

1 package sum;
2 import java.io.File;
3 import java.io.FileOutputStream;
4 import java.io.PrintWriter;
5 public class Sum extends Thread {
6     int sum = 0;
7     public void run() {
8         try {
9             File f = new File(pathname: "output.txt");
10            FileOutputStream fout = new FileOutputStream(file: f);
11            PrintWriter pout = new PrintWriter(out: fout);
12            pout.print(i: sum);
13            pout.flush(); fout.close();
14        } catch (Exception e) {}
15    }
16    public Sum(int sum) {
17        this.sum = sum;
18    }
19    public static void main(String[] args) {
20        try {
21            for (int i = 0; i < args.length; i++) {
22                Integer.parseInt(args[i]);
23            }
24        } catch (NumberFormatException e) {
25            System.out.println("error args");
26            System.exit(status: 0);
27        }
28        try {
29            int total=0;
30            for (int i = 0; i < args.length; i++) {
31                total += Integer.parseInt(args[i]);
32            }
33            Sum s = new Sum(sum: total);
34            s.start();
35        } catch (Exception e) {}
36    }
37 }

```

```

1 package badthread;
2 public class BankAccount {
3     volatile int money = 0;
4     Object o = new Object();
5     public BankAccount(int money) {
6         this.money = money;
7     }
8     public void deposit(int money) {
9         synchronized (o) {
10             for (int i = 0; i < money; i++) {
11                 this.money++;
12             }
13         }
14     }
15     public void withdraw(int money) {
16         synchronized (o) {
17             for (int i = 0; i < money; i++) {
18                 this.money--;
19             }
20         }
21     }
22     public int getBalance() {
23         return money;
24     }
25 }
26 }

```

```

1 package badthread;
2 public class BankBranch extends Thread {
3     BankAccount bankAcct = null;
4     String method = null;
5     int money = 0;
6     public BankBranch(BankAccount bankAcct, String method, int money) {
7         this.bankAcct = bankAcct;
8         this.method = method;
9         this.money = money;
10    }
11    public void deposit(int money) {
12        bankAcct.deposit(money);
13    }
14    public void withdraw(int money) {
15        bankAcct.withdraw(money);
16    }
17    public void run() {
18        if (method.equals(anObject: "deposit")) {
19            deposit(money);
20        } else {
21            withdraw(money);
22        }
23    }
24 }
25 }

```

```

JavaTextCopy.java
3 import java.io.BufferedReader;
4 import java.io.File;
5 import java.io.FileInputStream;
6 import java.io.FileOutputStream;
7 import java.io.InputStreamReader;
8 import java.io.PrintWriter;
9 public class JavaTextCopy {
10
11     public static void main(String[] args) {
12         if(args.length != 2){
13             System.out.println("Usage: java javaTextCopy <source file> <destination file>");
14         }
15         else{
16             try{
17                 String msg;
18                 File data1 = new File(args[0]);
19                 FileInputStream fin = new FileInputStream(data1);
20                 InputStreamReader ir = new InputStreamReader(fin);
21                 BufferedReader br = new BufferedReader(ir);
22
23                 File data2 = new File(args[1]);
24                 FileOutputStream fout = new FileOutputStream(data2);
25                 PrintWriter pout = new PrintWriter(fout);
26                 while ((msg=br.readLine()) != null){
27                     pout.println(msg);
28                 }
29                 br.close();
30                 pout.close();
31             }catch (Exception e) {
32                 e.printStackTrace();
33             }
34         }
35     }
36 }

```

```

JavaThread.java
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/
3 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/
4 */
5 package javathread;
6
7 /**
8  *
9  * @author thasa
10 */
11 public class JavaThread extends Thread{
12     int n;
13     public JavaThread(int n){
14         this.n = n;
15     }
16     public void run(){
17         System.out.println(n + " " + "Hello Word");
18     }
19
20     public static void main(String[] args) {
21         if(args.length != 1){
22             System.out.println("Please enter one number.");
23         }else{
24             try{
25                 int c=Integer.parseInt(args[0]);
26                 for(int i=0;i<c;i++){
27                     JavaThread t = new JavaThread(i);
28                     t.start();
29                 }
30             }catch(NumberFormatException e){}
31         }
32     }
33 }

```

```

JavaTwoThread 1.java
11 public class JavaTwoThread implements Runnable {
12
13     int from;
14     int to;
15     int sum = 0;
16
17     public JavaTwoThread(int from, int to) {
18         this.from = from;
19         this.to = to;
20     }
21
22     public void run() {
23         for (int i = from; i <= to; i++) {
24             sum += i;
25         }
26         try {
27             if (to == 5000) {
28                 Thread.sleep(5000);
29             } else {
30                 Thread.sleep(10000);
31             }
32         } catch (Exception e) {}
33     }
34
35     public int getSum() {
36         return sum;
37     }
38
39     public static void main(String[] args) {
40         JavaTwoThread j1 = new JavaTwoThread(1, 5000);
41         JavaTwoThread j2 = new JavaTwoThread(5001, 10000);
42         Thread t1 = new Thread(j1);
43         Thread t2 = new Thread(j2);
44         try {
45             t1.start();
46             t2.start();
47             t1.join();
48             t2.join();
49         } catch (Exception e) {}
50         System.out.println(j1.getSum() + j2.getSum());
51     }
52 }

```

```

FileAVG.java
2 import java.io.BufferedReader;
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.FileOutputStream;
6 import java.io.InputStreamReader;
7 import java.io.PrintWriter;
8 public class FileAVG extends Thread {
9     float avg;
10     public void run() {
11         try {
12             File f = new File(pathname: "output.txt");
13             FileOutputStream fout = new FileOutputStream(file: f);
14             PrintWriter pout = new PrintWriter(out: fout);
15             pout.print("AVG = " + avg);
16             pout.flush();
17             pout.close();
18         } catch (Exception e) {}
19     }
20     public FileAVG(float avg) {
21         this.avg = avg;
22     }
23     public static void main(String[] args) {
24         try {
25             File f = new File(pathname: "C:\\Users\\thasa\\Desktop\\networkprogram\\NewFolder\\FileAVG\\input.txt");
26             BufferedReader br = new BufferedReader(
27                 new InputStreamReader(
28                     new FileInputStream(file: f)));
29
30             String msg;
31             float avg = 0;
32             while ((msg = br.readLine()) != null) {
33                 try {
34                     int n = Integer.parseInt(msg);
35                     avg += n;
36                 } catch (Exception e) {}
37             }
38             FileAVG a = new FileAVG(avg / 2);
39             a.start();
40         } catch (Exception e) {}
41     }
42 }

```

2. จงเขียนโปรแกรมภาษา Java FileAVG.java โดยโปรแกรมจะอ่านค่าตัวเลขจากไฟล์ที่มีชื่อตามจาก
 ชาติวัฒนธรรม ซึ่งไฟล์ที่มีชื่อตัวเลขจะเป็นตัวเลขเต็ม 1 คำต่อ 1 บรรทัด จากนั้นโปรแกรมจะหาค่าเฉลี่ยของตัวเลข
 ที่รวมค่าที่ได้ (เฉพาะตัวเลขจำนวนเต็มเท่านั้น) แล้วนำค่าเฉลี่ยนั้นไปใส่ลงไฟล์ชื่อ c:\average.txt
 โปรแกรมจะทำงานจนจบชีวิต แล้วค่าเฉลี่ยที่ได้จะนำไปใส่ลงไฟล์ชื่อ c:\average.txt
 สำหรับ Error ที่อาจเกิดจากการไม่ใส่ชื่อไฟล์ตามที่กำหนดให้จะมีความเหมาะสม (10 คะแนน)

ตัวอย่าง	Java FileAVG c:\input.txt	ไฟล์ c:\output.txt ที่โปรแกรมสร้าง	คำอธิบาย
3.5	AVG=12.5	ไม่ไฟล์ output.txt มีเพียงแค่ บรรทัดที่มีเลข 4 และ 21 ที่ เป็นจำนวนเต็ม การหาค่าเฉลี่ยก็คือ (4 + 21) / 2 = 12.5	
4			
ABC			
21			

```

JavaSyncTest.java
11 public class JavaSyncTest implements Runnable {
12
13     static volatile int balance = 0;
14     static Object o = new Object();
15
16     public void run() {
17         for (int i = 0; i < 100000; i++) {
18             synchronized (o) {
19                 balance++;
20             }
21         }
22     }
23
24     public int getBalance(){
25         return balance;
26     }
27
28     public static void main(String[] args) {
29         JavaSyncTest j1 = new JavaSyncTest();
30         JavaSyncTest j2 = new JavaSyncTest();
31         JavaSyncTest j3 = new JavaSyncTest();
32
33         Thread t1 = new Thread(j1);
34         Thread t2 = new Thread(j2);
35         Thread t3 = new Thread(j3);
36
37         t1.start(); t2.start(); t3.start();
38         try {
39             t1.join();
40             t2.join();
41             t3.join();
42         } catch (Exception e) {}
43         System.out.println("Balance = " + j1.getBalance());
44     }
45 }

```

```

Grade.java
1 package grade;
2
3 public class Grade {
4
5     public static void main(String[] args) {
6         try {
7             if (args.length != 1) {
8                 System.out.println("กรุณาระบุคะแนนของคุณ");
9             } else if (Integer.parseInt(args[0]) < 0 || Integer.parseInt(args[0]) > 100) {
10                 System.out.println("กรุณาระบุค่าเลข 0-100");
11             } else {
12                 int grade = Integer.parseInt(args[0]);
13                 if (grade >= 80) {
14                     System.out.println("A");
15                 } else if (grade >= 70) {
16                     System.out.println("B");
17                 } else if (grade >= 60) {
18                     System.out.println("C");
19                 } else if (grade >= 50) {
20                     System.out.println("D");
21                 } else {
22                     System.out.println("F");
23                 }
24             }
25         } catch (NumberFormatException e) {
26             System.out.println("กรุณาระบุตัวเลขจำนวนเต็ม");
27         }
28     }
29 }
30
31
32

```

```

JavaList.java
1 package javalist;
2
3 import java.io.File;
4 import java.rmi.server.ExportException;
5
6 public class JavaList {
7
8     public static void main(String[] args) {
9         if (args.length != 1) {
10             System.out.println("Usage: java JavaList <File/Directory name>");
11         } else {
12             File f = new File(args[0]);
13             if (f.exists()) {
14                 if (f.isFile()) {
15                     System.out.println("File size = " + f.length());
16                 } else if (f.isDirectory()) {
17                     String[] ff = f.list();
18                     for (int i = 0; i < ff.length; i++) {
19                         System.out.println(ff[i]);
20                     }
21                 } else {
22                     System.out.println("BUG");
23                 }
24             } else {
25                 System.out.println("File not found");
26             }
27         }
28     }
29 }
30
31
32

```

```

TestArgs.java
1
2 package testargs;
3 public class TestArgs {
4     public static void main(String[] args) {
5         try {
6             if (args.length != 2) {
7                 System.out.println("Usage: java TestArgs <number1> <number2>");
8             } else {
9                 float num1 = Float.parseFloat(args[0]);
10                 float num2 = Float.parseFloat(args[1]);
11                 System.out.println("number = " + (num1*num2));
12             }
13         } catch (NumberFormatException e) {
14             System.out.println("Please enter your float.");
15         }
16     }
17 }
18

```

```

JavaBinaryCopy.java
Source History
1 package javabinarycopy;
2 import java.io.File;
3 import java.io.FileInputStream;
4 import java.io.FileOutputStream;
5 import java.io.IOException;
6 public class JavaBinaryCopy {
7     public static void main(String[] args) {
8         if (args.length != 2) {
9             System.out.println("Usage: java JavaBinaryCopy <source file> <destination file>");
10         } else {
11             try {
12                 File data = new File(args[0]);
13                 FileInputStream fin = new FileInputStream(data);
14                 byte[] a = new byte[16];
15
16                 File f = new File(args[1]);
17                 FileOutputStream fout = new FileOutputStream(f);
18
19                 int n;
20                 while ((n = fin.read(a)) > 0) {
21                     fout.write(a, 0, n);
22                 }
23                 fin.close();
24                 fout.close();
25             } catch (Exception e) {
26                 e.printStackTrace();
27             }
28         }
29     }
30 }
31
32
33

```

3. จงเขียนโปรแกรมภาษาจาวาในแฟ้มข้อมูลชื่อ Exam3.java ซึ่ง Class Exam3 จะเป็น Class ที่สามารถทำงานเป็น Thread ด้วยวิธี extends Thread (15 คะแนน)

- โปรแกรมจะรับอาร์กิวเมนต์ 1 ค่า คือ จำนวน thread ที่ต้องการสร้าง โดยจะส่ง thread ที่ถูกสร้างขึ้นนี้พิมพ์ข้อความว่า "Hello World" เป็นจำนวนครั้งตามลำดับที่ถูกสร้างขึ้นมา พร้อมทั้งจะกลับหลังจากพิมพ์ข้อความเสร็จเป็นจำนวน "วินาที" ตามลำดับที่ถูกสร้างขึ้นมา
- ถ้าผู้ใช้ป้อนจำนวนอาร์กิวเมนต์ไม่เท่ากับ 1 ค่า หรือค่าที่ป้อนเข้ามาไม่ใช่จำนวนเต็ม ให้แสดงคำว่า "Error argument" แล้วจบโปรแกรม

การใช้งานโปรแกรม: ถ้าผู้ใช้ป้อน argument เป็น 3 ให้กับโปรแกรม โปรแกรม (main) จะสร้าง thread ขึ้นมา 3 threads โดย thread ตัวแรกจะพิมพ์คำว่า "Hello World" 1 ครั้ง และกลับไป 1 วินาที, thread ตัวที่ 2 จะพิมพ์คำว่า "Hello World" 2 ครั้ง แล้วกลับไป 2 วินาที ส่วน thread ตัวสุดท้ายคือตัวที่ 3 จะพิมพ์คำว่า "Hello World" 3 ครั้ง แล้วกลับไป 3 วินาที ซึ่งการทำงานรวมของโปรแกรมจะใช้เวลาประมาณ 3 วินาที

```

Exam3.java
Source History
1 package exam3;
2 public class Exam3 extends Thread {
3     String s;
4     long sleep;
5     int count;
6     public void run() {
7         try {
8             for (int i = 0; i < count; i++) {
9                 System.out.println(s);
10             }
11             Thread.sleep(sleep);
12         } catch (Exception e) {}
13     }
14     public Exam3(String s, int sleep, int count) {
15         this.s = s;
16         this.sleep = sleep;
17         this.count = count;
18     }
19     public static void main(String[] args) {
20         if (args.length != 1) {
21             System.out.println("Error argument");
22             System.exit(0);
23         }
24         int n = Integer.parseInt(args[0]);
25         for (int i = 1; i <= n; i++) {
26             Exam3 e = new Exam3(s: "Hello World", i * 1000, count:i);
27             e.start();
28             try {
29                 e.join();
30             } catch (Exception ee) {}
31         }
32     }
33 }
34

```

66-030524-2109-7 : นางสาวทัศนวรรณ สอนองพัน		สถานภาพนักศึกษา : ปกติ
มอพ. กรุงเทพฯ	ระดับปริญญาตรีต่อเนื่อง 2 - 3 ปี	กลุ่มเรียน : ECT-DE-RC รอบเช้า
วิทยาลัยเทคโนโลยีอุตสาหกรรม	ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์	สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์(ECT)
หลักสูตรอุตสาหกรรมศาสตรบัณฑิต สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ (ต่อเนื่อง)		

ข้อมูลผลการเรียนอย่างไม่เป็นทางการ

ปีการศึกษา 1/2567									
รหัสวิชา		ชื่อวิชา			ตอนเรียน	หน่วยกิต	เกรด		
030523200		SPECIAL PROJECT I			1	1	A		
030523503		MOBILE APPL DEVELOPMENT			1	2	A		
030523504		ARTIFICIAL INTELLIGENCE			1	2	A		
030523505		CYBERSECURITY			1	2	B+		
030523603		MOBILE APPL DEVELOPMENT LAB			1	1	A		
030523604		ARTIFICIAL INTELLIGENCE LAB			1	1	A		
030523605		CYBERSECURITY LABORATORY			1	1	B+		
030523701		CLOUD COMPUTING			1	2	B+		
030523801		CLOUD COMPUTING LABORATORY			1	1	B+		
030923102		SCIENCES IN DAILY LIFE			1	3	A		
030933155		ENGL CONVERS FOR DAILY LIFE			4	3	B+		
030953115		MEDITATION FOR LIFE DEV			3	3	A		
					ผลการศึกษา : ปกติ				
THIS SEMESTER					CUMULATIVE TO THIS SEMESTER				
CA	CS	CG	GP	GPA	CA	CS	CG	GP	GPAX
22	22	22	83.5	3.79	64	64	64	223.5	3.49

หมายเหตุ

CA : จำนวนหน่วยกิตที่ลงทะเบียน

CS : จำนวนหน่วยกิตที่ผ่าน

CG : จำนวนหน่วยกิตที่คิดเกรด

GP : ค่าคะแนน

GPA : คะแนนเฉลี่ยรายภาค

GPAX : คะแนนเฉลี่ยสะสม

หมายเหตุ

- เกรดที่แสดง จะทำการคำนวณเกรดรายวิชา สะสมยอดต่อเนื่องเพื่อให้เห็นการเคลื่อนไหวของยอดแต้มคะแนน
- สำหรับรายการที่แสดงใน Transcript จะถือค่าที่ได้คำนวณเก็บไว้ ณ วันสิ้นภาค เป็นค่าที่พิมพ์ใน Transcript
- เกรดที่ยังไม่แสดง อยู่ระหว่างดำเนินงานตรวจสอบของเจ้าหน้าที่ประมวลผล หรือ รอผลจากอาจารย์ผู้สอน