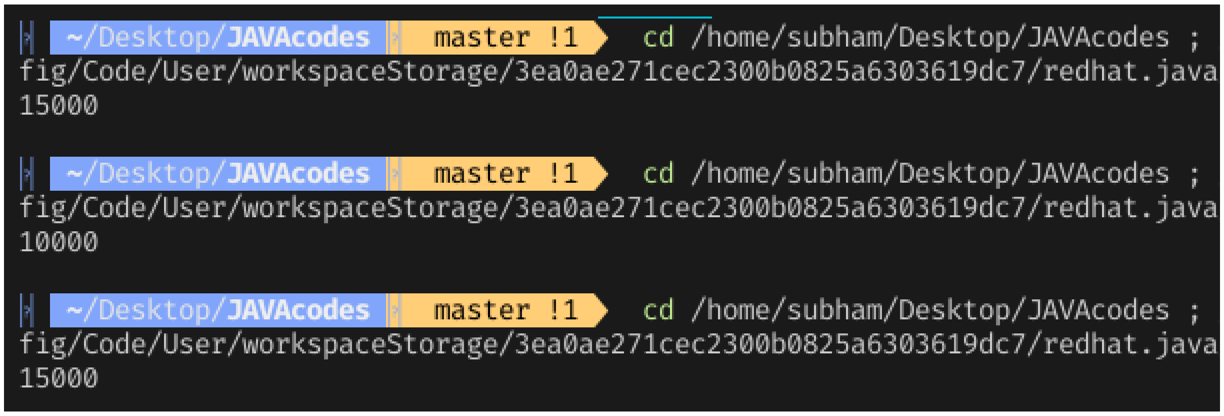
JAVA PROGRAMMING EXERCISE - APRIL 15th

1. Unsynchronized Threads

Code:

```java
public class ThreadUnsynchronized {
   public static void main(String[] args) throws Throwable {

       // UNSYNCHRONIZED THREADS
       account s = new account(20000);
       Thread thr1 = new Thread(new Runnable(){
           @Override
           public void run() {
               for(int i=0;i<50;i++) {
                   s.withdraw(100);
               }
           }
       });
       Thread thr2 = new Thread(new Runnable(){
           @Override
           public void run() {
               for(int i=0;i<50;i++) {
                   s.withdraw(100);
               }
           }
       });
       thr1.start();
       thr2.start();
       thr1.join();
       thr2.join();
       System.out.println(s.balance);
   }
}

class account {
   public int balance;

   public account(int deposit) {
       this.balance = deposit;
   }

   public void withdraw(int withdraw_amount) {
       this.balance = this.balance - withdraw_amount;
   }
}
```

Output:



```
 ~/Desktop/JAVAcodes   master !1   cd /home/subham/Desktop/JAVAcodes ;
fig/Code/User/workspaceStorage/3ea0ae271cec2300b0825a6303619dc7/redhat.java
15000

 ~/Desktop/JAVAcodes   master !1   cd /home/subham/Desktop/JAVAcodes ;
fig/Code/User/workspaceStorage/3ea0ae271cec2300b0825a6303619dc7/redhat.java
10000

 ~/Desktop/JAVAcodes   master !1   cd /home/subham/Desktop/JAVAcodes ;
fig/Code/User/workspaceStorage/3ea0ae271cec2300b0825a6303619dc7/redhat.java
15000
```

Since the threads are unsynchronized, the output of the account balance is different every time we run the output, and even the final balance is wrong sometimes.

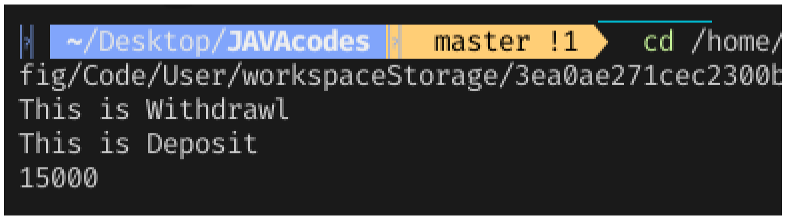
2. <u>Synchronized threads - Using synchronized functions:</u>

<u>Code:</u>

```java
public class ThreadSynchronized {
   public static void main(String[] args) throws Throwable {

       // SYNCHRONIZED THREADS
       bankAccount s = new bankAccount(20000);
       Thread thr1 = new Thread(new Runnable(){
           @Override
           public void run() {
               s.withdraw(10000);
           }
       });
       Thread thr2 = new Thread(new Runnable(){
           @Override
           public void run() {
               s.deposit(5000);
           }
       });
       thr1.start();
       thr2.start();
       thr1.join();
       thr2.join();
       System.out.println(s.balance);
   }
}

class bankAccount {
   public int balance;

   public bankAccount(int deposit) {
       this.balance = deposit;
   }

   public synchronized void withdraw(int withdraw_amount) {
       System.out.println("This is Withdrawl");
       this.balance = this.balance - withdraw_amount;
   }
```

```
    public synchronized void deposit(int deposit_amount) {
        System.out.println("This is Deposit");
        this.balance = this.balance + deposit_amount;
    }
}
```

<u>Output:</u>



```
 ~/Desktop/JAVAcodes   master !1   cd /home/
fig/Code/User/workspaceStorage/3ea0ae271cec2300b
This is Withdrawl
This is Deposit
15000
```

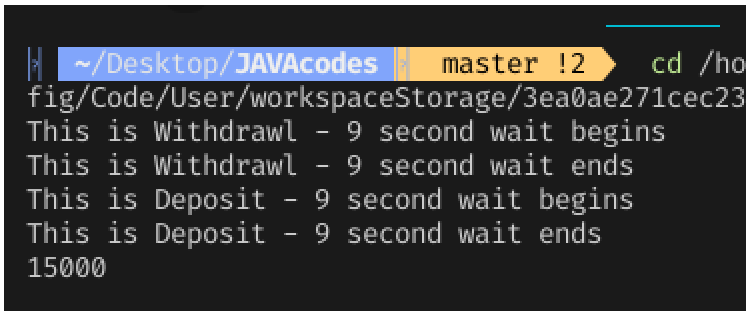
The code on the threads is now synchronized. The withdrawal runs first then the deposit and finally we get the correct final account balance.

3. <u>Synchronized Threads - Using sleep() to simulate time taken to run the withdraw and deposit function:</u>

<u>Code:</u>

```java
public class ThreadSynchronizedWithSleep {
   public static void main(String[] args) throws Throwable {

       // SYNCHRONIZED THREADS
       bankAccount1 s = new bankAccount1(20000);
       Thread thr1 = new Thread(new Runnable() {
           @Override
           public void run() {
               try {
                   s.withdraw(10000);
               } catch (Throwable e) {
                   e.printStackTrace();
               }
           }
       });
       Thread thr2 = new Thread(new Runnable() {
           @Override
           public void run() {
               try {
                   s.deposit(5000);
               } catch (Throwable e) {
                   e.printStackTrace();
               }
           }
       });
       thr1.start();
       thr2.start();
       thr1.join();
       thr2.join();
       System.out.println(s.balance);
   }
}

class bankAccount1 {
   public int balance;
   public bankAccount1(int deposit) {
        this.balance = deposit;
   }

  public synchronized void withdraw(int withdraw_amount) throws Throwable
{
        System.out.println("This is Withdrawl - 9 second wait begins");
        Thread.currentThread().sleep(9000);
        System.out.println("This is Withdrawl - 9 second wait ends");
        this.balance = this.balance - withdraw_amount;
   }

   public synchronized void deposit(int deposit_amount) throws Throwable {
        System.out.println("This is Deposit - 9 second wait begins");
        Thread.currentThread().sleep(9000);
        System.out.println("This is Deposit - 9 second wait ends");
        this.balance = this.balance + deposit_amount;
   }
}
```

Output:



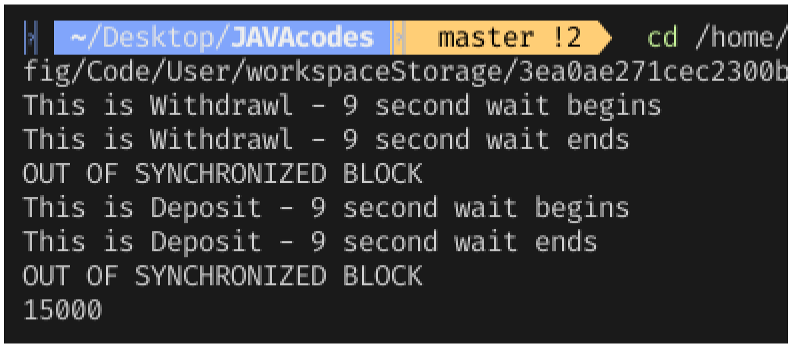
```
 ~/Desktop/JAVAcodes   master !2   cd /ho
fig/Code/User/workspaceStorage/3ea0ae271cec23
This is Withdrawl - 9 second wait begins
This is Withdrawl - 9 second wait ends
This is Deposit - 9 second wait begins
This is Deposit - 9 second wait ends
15000
```

4. Synchronized Threads - Using sleep() together with synchronized blocks:

Code:

```java
public class ThreadSynchronizedWithSleepAndSynchronizedBlocks {
    public static void main(String[] args) throws Throwable {

        // SYNCHRONIZED THREADS
        bankAccount1 s = new bankAccount1(20000);
        Thread thr1 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    s.withdraw(10000);
                } catch (Throwable e) {
                    e.printStackTrace();
                }
            }
        });
        Thread thr2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    s.deposit(5000);
                } catch (Throwable e) {
                    e.printStackTrace();
                }
            }
        });
        thr1.start();
        thr2.start();
        thr1.join();
        thr2.join();
        System.out.println(s.balance);
    }
}

class bankAccount1 {
    public int balance;
    public bankAccount1(int deposit) {
        this.balance = deposit;
    }

    public void withdraw(int withdraw_amount) throws Throwable {
        synchronized (this) {
            System.out.println("This is Withdrawl - 9 second wait begins");
            Thread.currentThread().sleep(9000);
            System.out.println("This is Withdrawl - 9 second wait ends");
            this.balance = this.balance - withdraw_amount;
        }
        System.out.println("OUT OF SYNCHRONIZED BLOCK");
    }

    public void deposit(int deposit_amount) throws Throwable {
        synchronized (this) {
            System.out.println("This is Deposit - 9 second wait begins");
            Thread.currentThread().sleep(9000);
            System.out.println("This is Deposit - 9 second wait ends");
            this.balance = this.balance + deposit_amount;

        }
        System.out.println("OUT OF SYNCHRONIZED BLOCK");
    }
}
```

Output:



```
  ~/Desktop/JAVAcodes    master !2    cd /home/
fig/Code/User/workspaceStorage/3ea0ae271cec2300b
This is Withdrawl - 9 second wait begins
This is Withdrawl - 9 second wait ends
OUT OF SYNCHRONIZED BLOCK
This is Deposit - 9 second wait begins
This is Deposit - 9 second wait ends
OUT OF SYNCHRONIZED BLOCK
15000
```

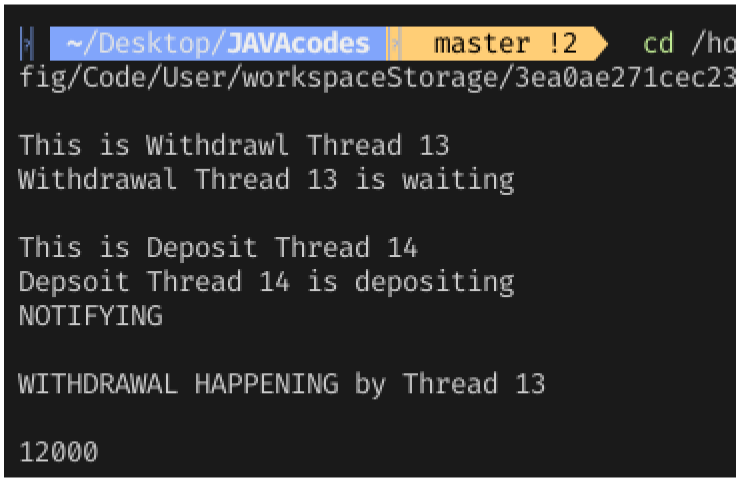
5. <u>Synchronized threads - Using wait() to make threads wait for a particular action, and notify() to notify one of the threads waiting:</u>

<u>Code:</u>

```
public class ThreadWaitNotify {
   public static void main(String[] args) throws Throwable {
       Account subham = new Account(2000);
       Thread thr1=new Thread(new Runnable(){
           @Override
           public void run() {
               try {
                   subham.withdraw(30000);
               } catch (Throwable e) {
                   // TODO Auto-generated catch block
                   e.printStackTrace();
               }
           }
       });
       Thread thr2=new Thread(new Runnable(){
           @Override
           public void run() {
               try {
                   subham.deposit(40000);
               } catch (Throwable e) {
                   // TODO Auto-generated catch block
                   e.printStackTrace();
               }
           }
       });
       thr1.start();
       thr2.start();
       thr1.join();
       thr2.join();
       System.out.println(subham.balance);
   }
}

class Account {
   int balance;
```

```
    public Account(int balance) {
        this.balance = balance;
    }

    public synchronized void withdraw(int withdraw_amount) throws Throwable
{
        System.out.println();
        System.out.println("This is Withdrawl Thread " +
Thread.currentThread().getId());
        while (withdraw_amount > balance) {
            System.out.println("Withdrawal Thread " +
Thread.currentThread().getId() + " is waiting");
            wait();
        }
        System.out.println("WITHDRAWAL HAPPENING by Thread "+
Thread.currentThread().getId());
        System.out.println();
        this.balance = this.balance - withdraw_amount;
    }

    public synchronized void deposit(int deposit_amount) throws Throwable {
        System.out.println();
        System.out.println("This is Deposit Thread
"+Thread.currentThread().getId());
        System.out.println("Depsoit Thread " +
Thread.currentThread().getId() + " is depositing");
        System.out.println("NOTIFYING");
        System.out.println();
        this.balance = this.balance + deposit_amount;
        notify();
    }
}
```

<u>Output:</u>



```
 ~/Desktop/JAVAcodes   master !2   cd /ho
fig/Code/User/workspaceStorage/3ea0ae271cec23

This is Withdrawl Thread 13
Withdrawal Thread 13 is waiting

This is Deposit Thread 14
Depsoit Thread 14 is depositing
NOTIFYING

WITHDRAWAL HAPPENING by Thread 13

12000
```

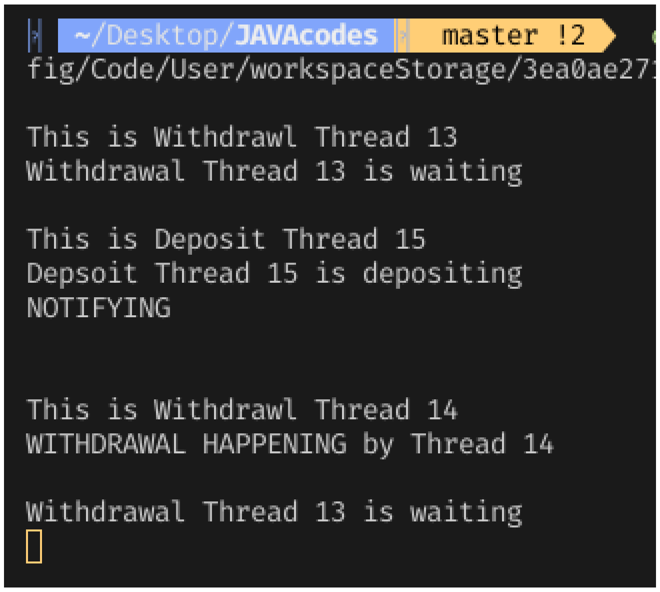
6. <u>Synchronized threads - Using wait() to make threads wait for a particular action, and notifyAll() to notify all of the threads waiting:</u>

<u>Code:</u>

```
public class ThreadWaitNotifyAll {
   public static void main(String[] args) throws Throwable {
       BankAccount subham = new BankAccount(2000);
       Thread thr1=new Thread(new Runnable(){
           @Override
           public void run() {
               try {
                   subham.withdraw(30000);
               } catch (Throwable e) {
                   // TODO Auto-generated catch block
                   e.printStackTrace();
               }
           }
       });
       Thread thr2=new Thread(new Runnable(){
           @Override
           public void run() {
               try {
                   subham.withdraw(40000);
               } catch (Throwable e) {
                   // TODO Auto-generated catch block
                   e.printStackTrace();
               }
           }
       });
       Thread thr3=new Thread(new Runnable(){
           @Override
           public void run() {
               try {
                   subham.deposit(40000);
               } catch (Throwable e) {
                   // TODO Auto-generated catch block
                   e.printStackTrace();
               }
           }
       });
       thr1.start();
       thr2.start();
       thr3.start();
       thr1.join();
       thr2.join();
       thr3.join();
       System.out.println(subham.balance);
   }
}

class BankAccount {
   int balance;

   public BankAccount(int balance) {
       this.balance = balance;
   }

   public synchronized void withdraw(int withdraw_amount) throws Throwable
{
       System.out.println();
       System.out.println("This is Withdrawl Thread " +
Thread.currentThread().getId());
       while (withdraw_amount > balance) {
           System.out.println("Withdrawal Thread " +
Thread.currentThread().getId() + " is waiting");
           wait();
       }
       System.out.println("WITHDRAWAL HAPPENING by Thread "+
Thread.currentThread().getId());
       System.out.println();
       this.balance = this.balance - withdraw_amount;
   }

   public synchronized void deposit(int deposit_amount) throws Throwable {
       System.out.println();
       System.out.println("This is Deposit Thread
"+Thread.currentThread().getId());
```

```
        System.out.println("Depsoit Thread " +
Thread.currentThread().getId() + " is depositing");
        System.out.println("NOTIFYING");
        System.out.println();
        this.balance = this.balance + deposit_amount;
        notifyAll();
    }
}
```

Output:



```
 ~/Desktop/JAVAcodes   master !2
fig/Code/User/workspaceStorage/3ea0ae271

This is Withdrawl Thread 13
Withdrawal Thread 13 is waiting

This is Deposit Thread 15
Depsoit Thread 15 is depositing
NOTIFYING


This is Withdrawl Thread 14
WITHDRAWAL HAPPENING by Thread 14

Withdrawal Thread 13 is waiting
```

In this case the Withdrawal thread 13 is waiting since there is not enough balance to withdraw and it will keep waiting until the user deposits money and notifies all the waiting withdrawing threads and there is enough money to withdraw the given amount.

JAVA PROGRAMMING EXERCISE - APRIL 29th

Question 1:

a)

Any exception must be a subclass of the ***Throwable*** class in order to be able to be thrown. Since the user defined exception class in this case, i.e ***satishexception*** does not extend ***Throwable*** or a subclass of ***Throwable*** class such as the ***Exception*** class, it cannot be thrown as shown in the question, i.e.

throw new satishexception("I am userdefined exception");

The ***satishexception*** class in this case in implementing the ***Runnable*** interface which is used during multithreading functions.

Hence we need to extend the class ***satishexception*** from the class ***Throwable*** or a subclass of ***Throwable*** class such as the ***Exception*** class.

Another issue with the given code is that the given constructor of ***satishexcpetion*** class does not take any parameter as input. It needs to take a string parameter S as input which will then be used to to invoke the constructor of the parent ***Exception*** class using the ***super()*** method. So we also need to make the constructor parameterized.

The correct code for the user defined exception class would be:

```
class satishexception extends Exception {
   public satishexception(String S) {
       super(S);
   }
}
```

b)

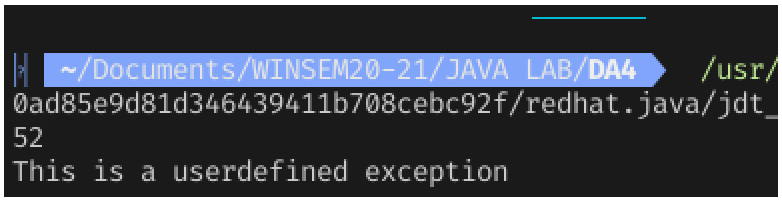
super(S) calls the constructor of the ***Exception*** class which in turn calls the constructor of the ***Throwable*** class where it sets the ***detailMessage*** of the ***Throwable*** class to S, and it is the same message which we can display using the ***getMessage()*** method on catching the exception.

Demonstration with code on how S can be used in catch block:

Code:

```
import java.util.Scanner;

public class qs1 {
   public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       try {
           int n = sc.nextInt();
           if(n > 10) {
               throw new satishexception("This is a userdefined
exception");
           }
       } catch (satishexception e) {
           System.out.println(e.getMessage());
       } finally {
           sc.close();
       }
   }
}

class satishexception extends Exception {
   public satishexception(String S) {
       super(S);
   }
}
```

Output:



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4    /usr/
0ad85e9d81d346439411b708cebc92f/redhat.java/jdt_
52
This is a userdefined exception
```

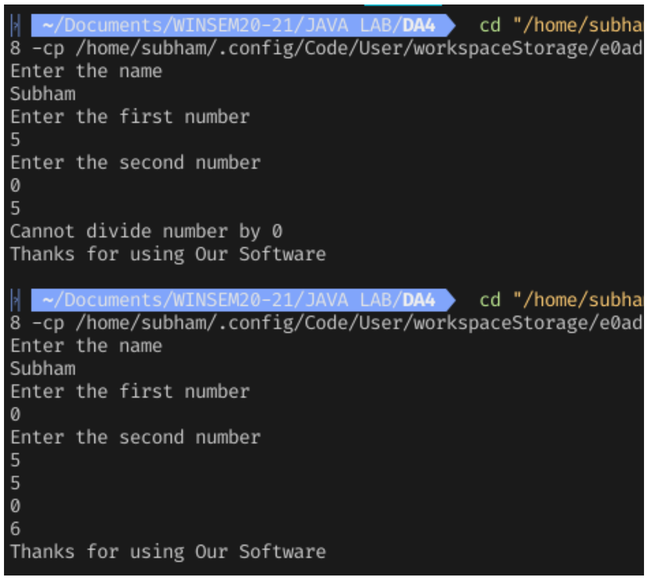
Question 2:

Code:

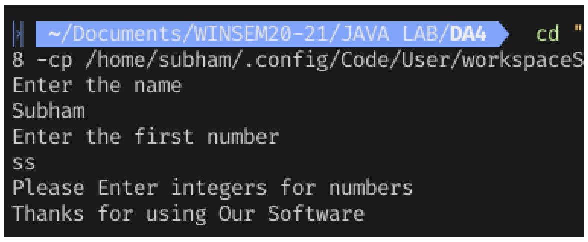
```java
import java.util.InputMismatchException;
import java.util.Scanner;

public class qs2 {
   public static void main(String[] args) {
       try {
           calculator c = new calculator();
           c.add();
           c.divide();
           c.display_namelength();
       } catch (NullPointerException e) {
           System.out.println(e.getMessage());
       } catch (InputMismatchException e) {
           System.out.println("Please Enter integers for numbers");
       } catch (ArithmeticException e) {
           System.out.println(e.getMessage());
       } catch (Exception e) {
           System.out.println(e.getMessage());
       } finally {
           System.out.println("Thanks for using Our Software");
       }

   }
}

class calculator {
   String name;
   int num1;
   int num2;

   public calculator() {
       Scanner input = new Scanner(System.in);
       System.out.println("Enter the name");
       this.name = input.nextLine();
       System.out.println("Enter the first number");
       this.num1 = input.nextInt();
```

```
        System.out.println("Enter the second number");
        this.num2 = input.nextInt();
        input.close();

    }

    public void add() {
        System.out.println(num1 + num2);
    }

    public void divide() throws ArithmeticException {
        if (num2 == 0)
            throw new ArithmeticException("Cannot divide number by 0");
        System.out.println(num1 / num2);
    }

    public void display_namelength() {
        System.out.println(name.length());
    }
}
```

Output:



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subha
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad
Enter the name
Subham
Enter the first number
5
Enter the second number
0
5
Cannot divide number by 0
Thanks for using Our Software

 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subha
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad
Enter the name
Subham
Enter the first number
0
Enter the second number
5
5
0
6
Thanks for using Our Software
```



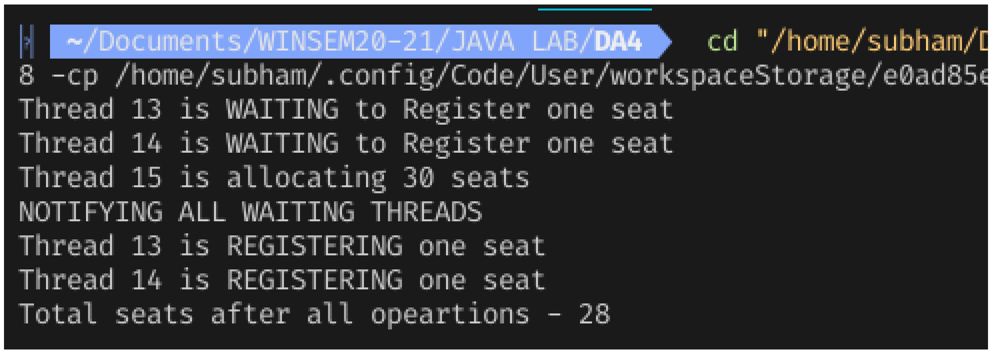
```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "
8 -cp /home/subham/.config/Code/User/workspaceS
Enter the name
Subham
Enter the first number
ss
Please Enter integers for numbers
Thanks for using Our Software
```

Question 3:

```
public class qs3 {
   public static void main(String[] args) throws Throwable {
       CourseRegistration c = new CourseRegistration();
       Thread thread1 = new Thread((Runnable) () -> {
           try {
               c.Register_seat();
           } catch (Throwable e) {
               System.out.println(e.getMessage());
           }
       });
       Thread thread2 = new Thread((Runnable) () -> {
           try {
               c.Register_seat();
           } catch (Throwable e) {
               System.out.println(e.getMessage());
           }
       });
       Thread thread3 = new Thread((Runnable) () -> {
           try {
               c.Allot_Seats(30);
           } catch (Throwable e) {
               System.out.println(e.getMessage());
           }
       });
       Thread thread4 = new Thread((Runnable) () -> {
           System.out.println("Total seats after all opeartions -
"+c.NumberOfSeats);
       });
       thread1.start();
       thread2.start();
       thread3.start();
       thread1.join();
       thread2.join();
       thread3.join();
       thread4.start();
   }
}
```

```java
class CourseRegistration {
    String CourseName;
    String FacultyName;
    int NumberOfSeats;

    CourseRegistration() {
        CourseName = "Java Programming";
        FacultyName = "Satish";
        NumberOfSeats = 0;
    }

    CourseRegistration(String CourseName, String FacultyName, int
NumberOfSeats) {
        this.CourseName = CourseName;
        this.FacultyName = FacultyName;
        this.NumberOfSeats = NumberOfSeats;
    }

    public synchronized void Register_seat() throws Throwable {
        while (!(NumberOfSeats > 0)) {
            System.out.println("Thread " + Thread.currentThread().getId() +
" is WAITING to Register one seat");
            wait();
        }
        System.out.println("Thread " + Thread.currentThread().getId() + "
is REGISTERING one seat");
        NumberOfSeats--;
    }

    public synchronized void Allot_Seats(int Seats) throws Throwable {
        System.out.println("Thread " + Thread.currentThread().getId() + "
is allocating " + Seats + " seats");
        NumberOfSeats = Seats;
        System.out.println("NOTIFYING ALL WAITING THREADS");
        notifyAll();
    }
}
```

Output:



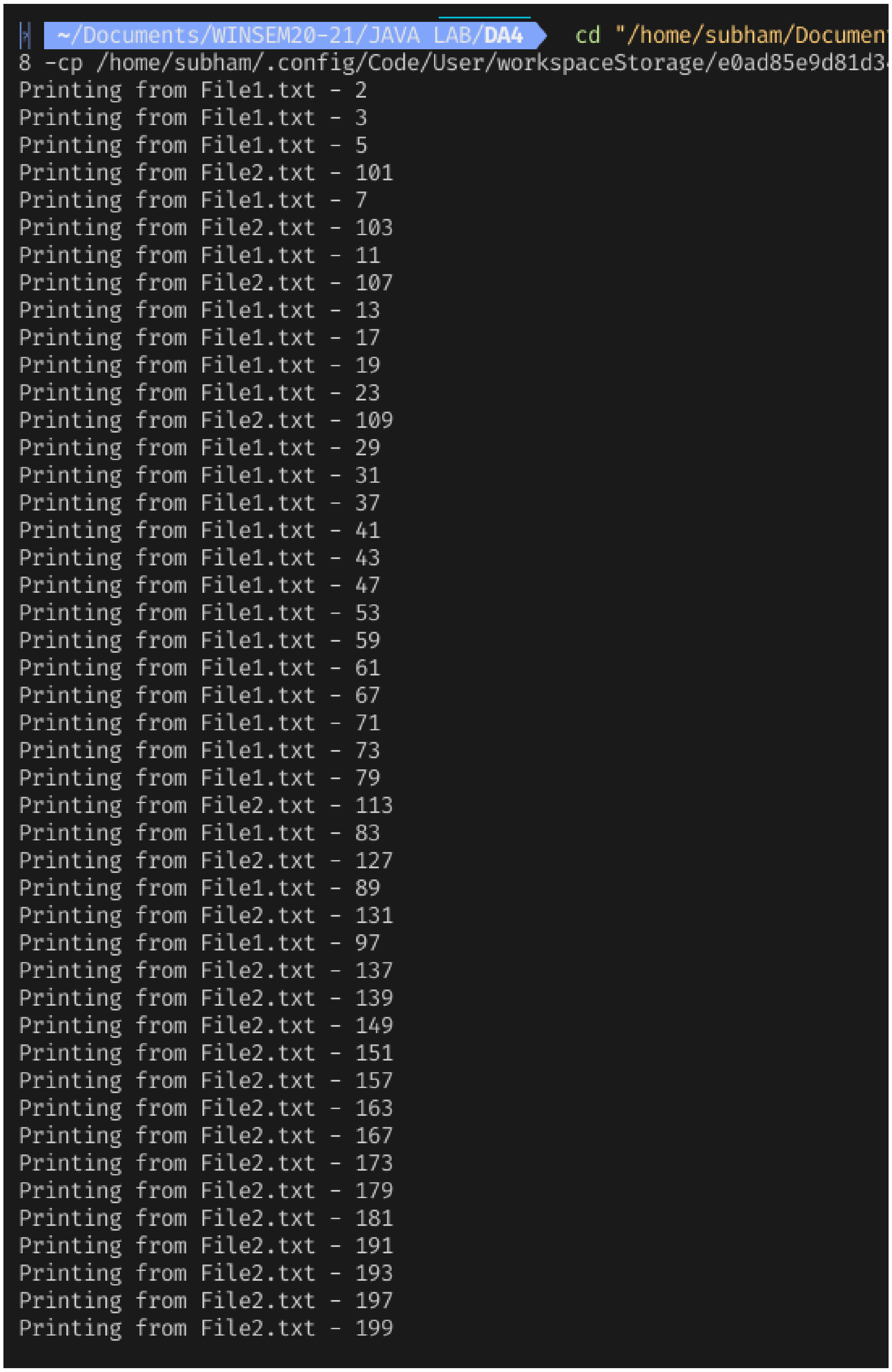
```
  ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subham/D
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad85e
Thread 13 is WAITING to Register one seat
Thread 14 is WAITING to Register one seat
Thread 15 is allocating 30 seats
NOTIFYING ALL WAITING THREADS
Thread 13 is REGISTERING one seat
Thread 14 is REGISTERING one seat
Total seats after all opeartions - 28
```

Question 4:

Code:

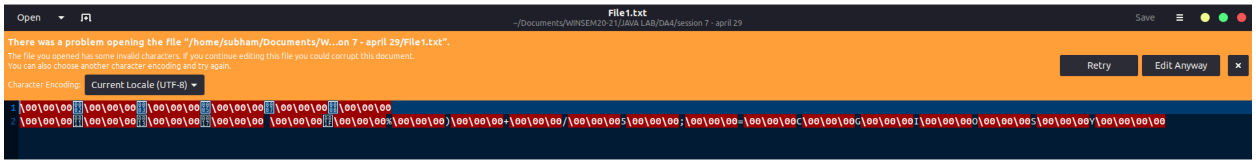
```
import java.io.*;

public class qs4 {
   public static void main(String[] args) throws Throwable {
       Thread thread1 = new Thread((Runnable) () -> {
           try {
               File obj = new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/File1.txt");
               FileOutputStream fout = new FileOutputStream(obj);
               DataOutputStream dout = new DataOutputStream(fout);
               for (int i = 2; i <= 100; i++) {
                   boolean isPrime = true;
                   for (int j = 2; j <= i / 2; j++) {
                       if (i % j == 0) {
                           isPrime = false;
                           break;
                       }
                   }
                   if (isPrime) {
                       dout.writeInt(i);
                       Thread.sleep(2000);
                   }
               }
               dout.close();
               fout.close();
           } catch (Exception e) {
               System.out.println(e.getMessage());
           }

       });

       Thread thread2 = new Thread((Runnable) () -> {
           try {
               File obj = new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/File2.txt");
               FileOutputStream fout = new FileOutputStream(obj);
               DataOutputStream dout = new DataOutputStream(fout);
               for (int i = 101; i <= 200; i++) {
                   boolean isPrime = true;
                   for (int j = 2; j <= i / 2; j++) {
                       if (i % j == 0) {
                           isPrime = false;
                           break;
                       }
                   }
                   if (isPrime) {
                       dout.writeInt(i);
                       Thread.sleep(2000);
                   }
               }
               dout.close();
               fout.close();
           } catch (Exception e) {
               System.out.println(e.getMessage());
           }
       });

       Thread thread3 = new Thread((Runnable) () -> {
           try {
               File obj = new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/File1.txt");
               FileInputStream fin = new FileInputStream(obj);
               DataInputStream din = new DataInputStream(fin);
               while (din.available() > 0) {
                   System.out.println("Printing from File1.txt - " +
din.readInt());
               }
               din.close();
               fin.close();
           } catch (Exception e) {
               System.out.println(e.getMessage());
           }
       });

       Thread thread4 = new Thread((Runnable) () -> {
           try {
               File obj = new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/File2.txt");
               FileInputStream fin = new FileInputStream(obj);
               DataInputStream din = new DataInputStream(fin);
               while (din.available() > 0) {
                   System.out.println("Printing from File2.txt - " +
din.readInt());
               }
               din.close();
               fin.close();
           } catch (Exception e) {
               System.out.println(e.getMessage());
           }
       });

       thread1.start();
       thread2.start();
       thread1.join();
       thread2.join();
       thread3.start();
       thread4.start();

   }
}
```

Output:



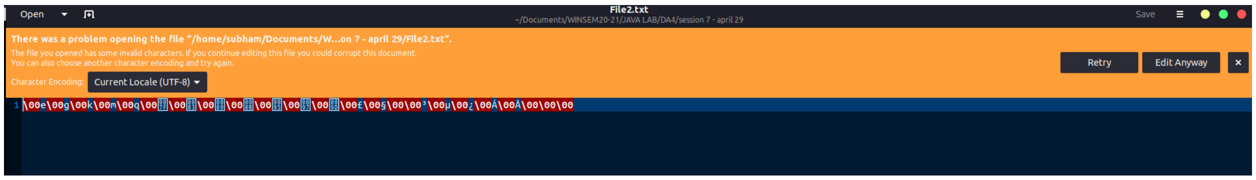
```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subham/Documen
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad85e9d81d3
Printing from File1.txt - 2
Printing from File1.txt - 3
Printing from File1.txt - 5
Printing from File2.txt - 101
Printing from File1.txt - 7
Printing from File2.txt - 103
Printing from File1.txt - 11
Printing from File2.txt - 107
Printing from File1.txt - 13
Printing from File1.txt - 17
Printing from File1.txt - 19
Printing from File1.txt - 23
Printing from File2.txt - 109
Printing from File1.txt - 29
Printing from File1.txt - 31
Printing from File1.txt - 37
Printing from File1.txt - 41
Printing from File1.txt - 43
Printing from File1.txt - 47
Printing from File1.txt - 53
Printing from File1.txt - 59
Printing from File1.txt - 61
Printing from File1.txt - 67
Printing from File1.txt - 71
Printing from File1.txt - 73
Printing from File1.txt - 79
Printing from File2.txt - 113
Printing from File1.txt - 83
Printing from File2.txt - 127
Printing from File1.txt - 89
Printing from File2.txt - 131
Printing from File1.txt - 97
Printing from File2.txt - 137
Printing from File2.txt - 139
Printing from File2.txt - 149
Printing from File2.txt - 151
Printing from File2.txt - 157
Printing from File2.txt - 163
Printing from File2.txt - 167
Printing from File2.txt - 173
Printing from File2.txt - 179
Printing from File2.txt - 181
Printing from File2.txt - 191
Printing from File2.txt - 193
Printing from File2.txt - 197
Printing from File2.txt - 199
```

<u>File1.txt:(unable to view specific encoding)</u>



```
File1.txt
~/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april 29

There was a problem opening the file "/home/subham/Documents/W...on 7 - april 29/File1.txt".
The file you opened has some invalid characters. If you continue editing this file you could corrupt this document.
You can also choose another character encoding and try again.
Character Encoding: Current Locale (UTF-8)

\00\00\00[02]\00\00\00[03]\00\00\00[05]\00\00\00[07]\00\00\00[0B]\00\00\00
\00\00\00[11]\00\00\00[13]\00\00\00[17]\00\00\00 \00\00\00[1F]\00\00\00%\00\00\00)\00\00\00+\00\00\00/\00\00\005\00\00\00;\00\00\00=\00\00\00C\00\00\00G\00\00\00I\00\00\00O\00\00\00S\00\00\00Y\00\00\00\00
```

<u>File2.txt:(unable to view specific encoding)</u>



```
File2.txt
~/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april 29

There was a problem opening the file "/home/subham/Documents/W...on 7 - april 29/File2.txt".
The file you opened has some invalid characters. If you continue editing this file you could corrupt this document.
You can also choose another character encoding and try again.
Character Encoding: Current Locale (UTF-8)

\00e\00g\00k\00m\00q\00[7F]\00[83]\00[89]\00[8B]\00[95]\00[97]\00[9D]\00£\00§\00\00³\00µ\00¿\00Á\00Å\00\00\00
```

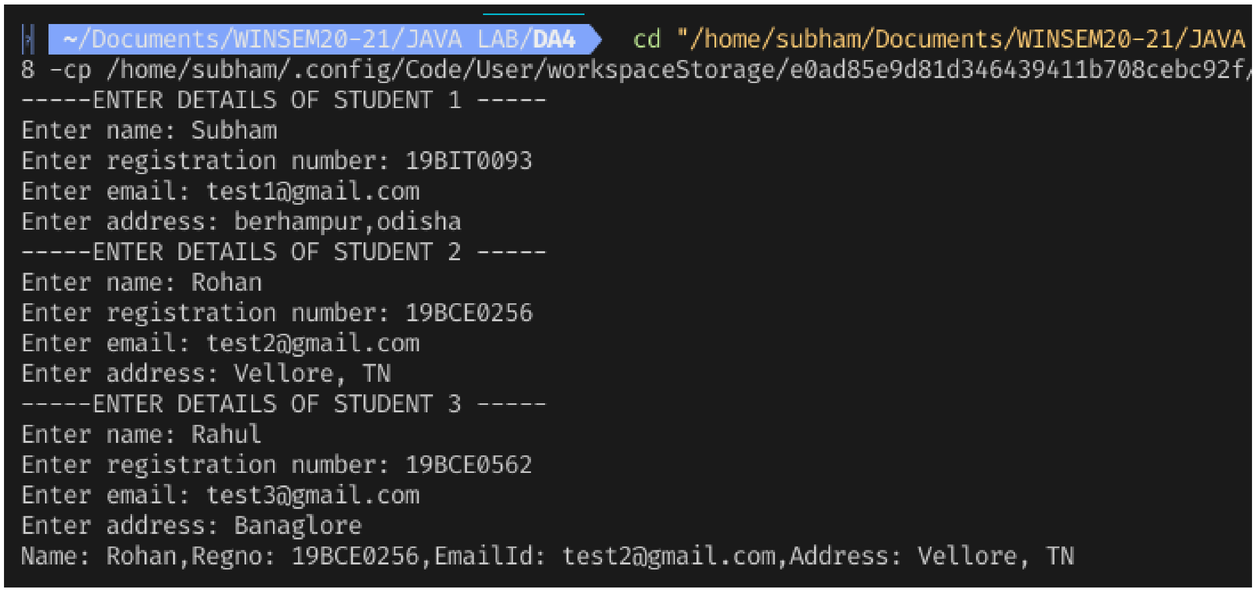
Question 5:

Code:

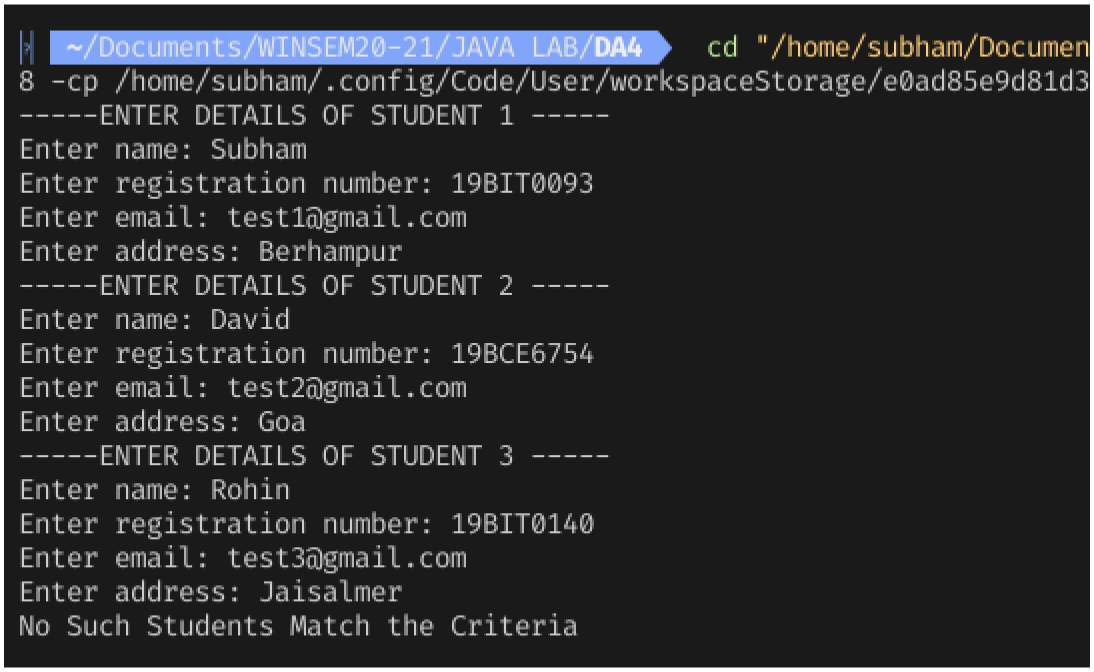
```
import java.io.*;
import java.util.Scanner;

public class qs5 {
   public static void main(String[] args) throws Throwable {
       Scanner sc = new Scanner(System.in);
       int num = 3;
       student sOutArr[] = new student[num];
       File obj = new File("/home/subham/Documents/WINSEM20-21/JAVA
LAB/DA4/session 7 - april 29/student.txt");
       FileOutputStream fout = new FileOutputStream(obj);
       ObjectOutputStream objout = new ObjectOutputStream(fout);
       for (int i = 0; i < num; i++) {
           System.out.println("-----ENTER DETAILS OF STUDENT " + (i + 1) +
" -----");
           System.out.print("Enter name: ");
           String name = sc.nextLine();
           System.out.print("Enter registration number: ");
           String regno = sc.nextLine();
           System.out.print("Enter email: ");
           String emailid = sc.nextLine();
           System.out.print("Enter address: ");
           String address = sc.nextLine();
           sOutArr[i] = new student(name, regno, emailid, address);
           objout.writeObject(sOutArr[i]);
       }
       objout.close();
       fout.close();
       sc.close();

       FileInputStream fin = new FileInputStream(obj);
       ObjectInputStream objin = new ObjectInputStream(fin);
       student sInpArr[] = new student[num];
       for (int i = 0; i < num; i++) {
           sInpArr[i] = (student) objin.readObject();
       }
```

```
        objin.close();
        fin.close();
        int flag = 0;
        for (student s : sInpArr) {
            if (s.regno.substring(2, 5).toLowerCase().compareTo("bce")==0
&& s.address.toLowerCase().contains("vellore")) {
                flag = 1;
                s.displayinfo();
            }
        }
        if (flag == 0) {
            System.out.println("No Such Students Match the Criteria");
        }

    }
}

class student implements Serializable {
    String name;
    String regno;
    String emailid;
    String address;

    student(String name, String regno, String emailid, String address) {
        this.name = name;
        this.regno = regno;
        this.emailid = emailid;
        this.address = address;
    }

    public void displayinfo() {
        System.out.println("Name: " + name + ",Regno: " + regno +
",EmailId: " + emailid + ",Address: " + address);
    }
}
```

Output:

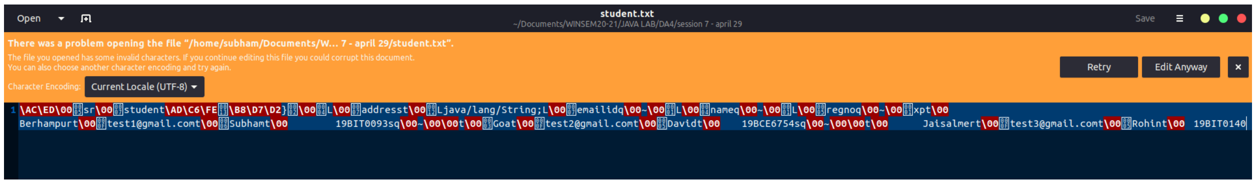


```
~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subham/Documents/WINSEM20-21/JAVA
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad85e9d81d346439411b708cebc92f/
-----ENTER DETAILS OF STUDENT 1 -----
Enter name: Subham
Enter registration number: 19BIT0093
Enter email: test1@gmail.com
Enter address: berhampur,odisha
-----ENTER DETAILS OF STUDENT 2 -----
Enter name: Rohan
Enter registration number: 19BCE0256
Enter email: test2@gmail.com
Enter address: Vellore, TN
-----ENTER DETAILS OF STUDENT 3 -----
Enter name: Rahul
Enter registration number: 19BCE0562
Enter email: test3@gmail.com
Enter address: Banaglore
Name: Rohan,Regno: 19BCE0256,EmailId: test2@gmail.com,Address: Vellore, TN
```



```
~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subham/Documen
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad85e9d81d3
-----ENTER DETAILS OF STUDENT 1 -----
Enter name: Subham
Enter registration number: 19BIT0093
Enter email: test1@gmail.com
Enter address: Berhampur
-----ENTER DETAILS OF STUDENT 2 -----
Enter name: David
Enter registration number: 19BCE6754
Enter email: test2@gmail.com
Enter address: Goa
-----ENTER DETAILS OF STUDENT 3 -----
Enter name: Rohin
Enter registration number: 19BIT0140
Enter email: test3@gmail.com
Enter address: Jaisalmer
No Such Students Match the Criteria
```

Student.txt:(unable to view specific encoding)



student.txt
~/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april 29

There was a problem opening the file "/home/subham/Documents/W... 7 - april 29/student.txt".
The file you opened has some invalid characters. If you continue editing this file you could corrupt this document.
You can also choose another character encoding and try again.

Character Encoding: Current Locale (UTF-8)

```
\AC\ED\00▯sr\00▯student\AD\C6\FE▯\B8\D7\D2}▯\00▯L\00▯addresst\00▯Ljava/lang/String;L\00▯emailidq\00~\00▯L\00▯nameq\00~\00▯L\00▯regnoq\00~\00▯xpt\00
Berhampurt\00▯test1@gmail.comt\00▯Subhamt\00        19BIT0093sq\00~\00\00t\00▯Goat\00▯test2@gmail.comt\00▯Davidt\00    19BCE6754sq\00~\00\00t\00      Jaisalmert\00▯test3@gmail.comt\00▯Rohint\00  19BIT0140
```

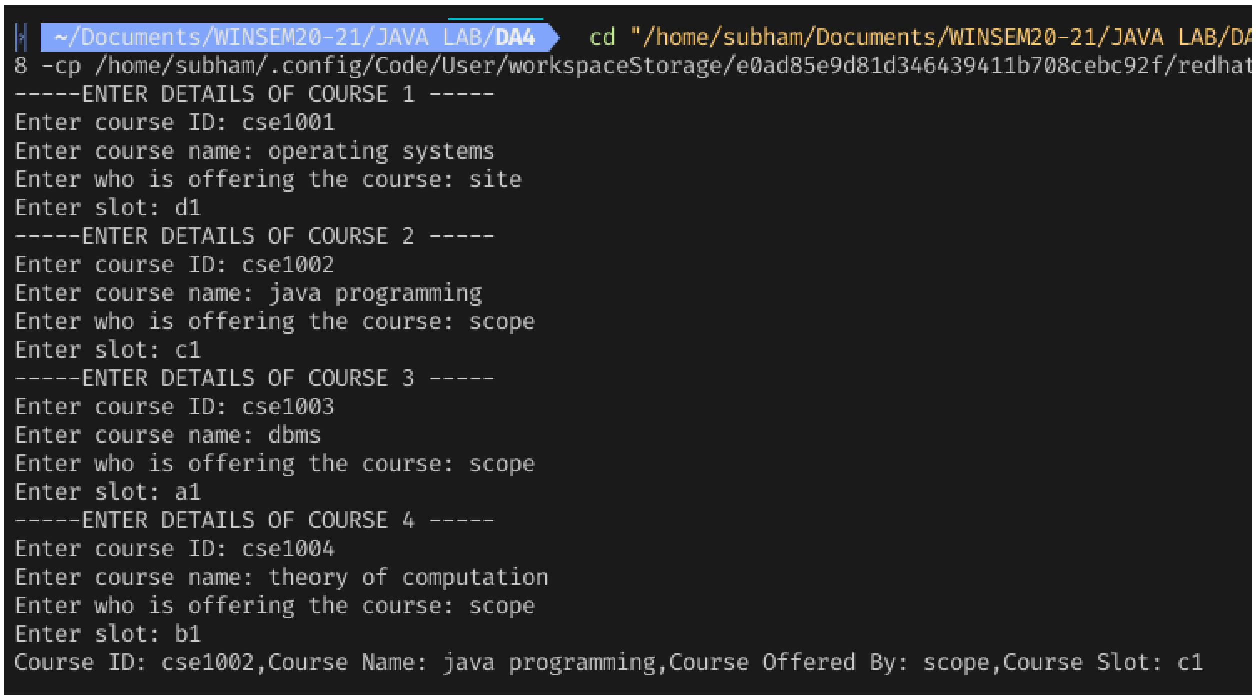
Question 6:

Code:

```
import java.io.*;
import java.util.Scanner;

public class qs6 {
   public static void main(String[] args) throws Throwable {
       Scanner sc = new Scanner(System.in);
       int num = 4;
       course cOutArr[] = new course[num];
       File obj = new File("/home/subham/Documents/WINSEM20-21/JAVA
LAB/DA4/session 7 - april 29/course.txt");
       FileOutputStream fout = new FileOutputStream(obj);
       ObjectOutputStream objout = new ObjectOutputStream(fout);
       for (int i = 0; i < num; i++) {
           System.out.println("-----ENTER DETAILS OF COURSE " + (i + 1) +
" -----");
           System.out.print("Enter course ID: ");
           String courseID = sc.nextLine();
           System.out.print("Enter course name: ");
           String courseName = sc.nextLine();
           System.out.print("Enter who is offering the course: ");
           String courseOfferedBy = sc.nextLine();
           System.out.print("Enter slot: ");
           String Course_slot = sc.nextLine();
           cOutArr[i] = new course(courseID, courseName, courseOfferedBy,
Course_slot);
           objout.writeObject(cOutArr[i]);
       }
       objout.close();
       fout.close();
       sc.close();

       FileInputStream fin = new FileInputStream(obj);
       ObjectInputStream objin = new ObjectInputStream(fin);
       course cInpArr[] = new course[num];
       for (int i = 0; i < num; i++) {
           cInpArr[i] = (course) objin.readObject();
```

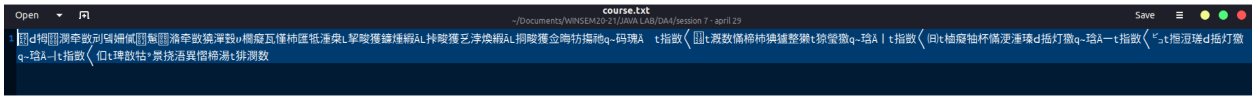
```
        }
        objin.close();
        fin.close();
        int flag = 0;
        for (course c : cInpArr) {
            if (c.courseName.compareToIgnoreCase("java programming")==0 &&
c.Course_slot.compareToIgnoreCase("c1")==0 &&
c.courseOfferedBy.compareToIgnoreCase("scope") == 0) {
                flag = 1;
                c.displayinfo();
            }
        }
        if (flag == 0) {
            System.out.println("No Such Courses Match the Criteria");
        }

    }
}

class course implements Serializable {
    String courseID;
    String courseName;
    String courseOfferedBy;
    String Course_slot;

    course(String courseID, String courseName, String courseOfferedBy,
String Course_slot) {
        this.courseID = courseID;
        this.courseName = courseName;
        this.courseOfferedBy = courseOfferedBy;
        this.Course_slot = Course_slot;
    }

    public void displayinfo() {
        System.out.println("Course ID: " + courseID + ",Course Name: " +
courseName + ",Course Offered By: " + courseOfferedBy + ",Course Slot: " +
Course_slot);
    }
}
```

Output:



```
~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subham/Documents/WINSEM20-21/JAVA LAB/DA
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ad85e9d81d346439411b708cebc92f/redhat
-----ENTER DETAILS OF COURSE 1 -----
Enter course ID: cse1001
Enter course name: operating systems
Enter who is offering the course: site
Enter slot: d1
-----ENTER DETAILS OF COURSE 2 -----
Enter course ID: cse1002
Enter course name: java programming
Enter who is offering the course: scope
Enter slot: c1
-----ENTER DETAILS OF COURSE 3 -----
Enter course ID: cse1003
Enter course name: dbms
Enter who is offering the course: scope
Enter slot: a1
-----ENTER DETAILS OF COURSE 4 -----
Enter course ID: cse1004
Enter course name: theory of computation
Enter who is offering the course: scope
Enter slot: b1
Course ID: cse1002,Course Name: java programming,Course Offered By: scope,Course Slot: c1
```

Course.txt:(unable to view specific encoding)



```
d栂[illegible]潤牵敳刋딕姍偑[illegible]鬤[illegible]潸牵敳獍湋毀ง檲癡瓦慬杮匯牴湩奔L挐晙獲鑤爄緞ĀL拵晙獲乭浡ǀ L搁睑獲佥晦敲䡂祑q~码瑰Ā  t指散〈[illegible]t澱数慉栝柹獭瀘整獺t獂瑩獭q~玲Ā丨t指散〈㈤t楂癡柚杯慉浬瀍漶瑧d掻灯瑳q~玲Ā一t指散〈ᄇ₃t挕洠瑳d掻灯瑳
q~玲Ā—|t指散〈㐾t琠敀牲ᵒ景挠浢慉栝湬t犇潤数
```

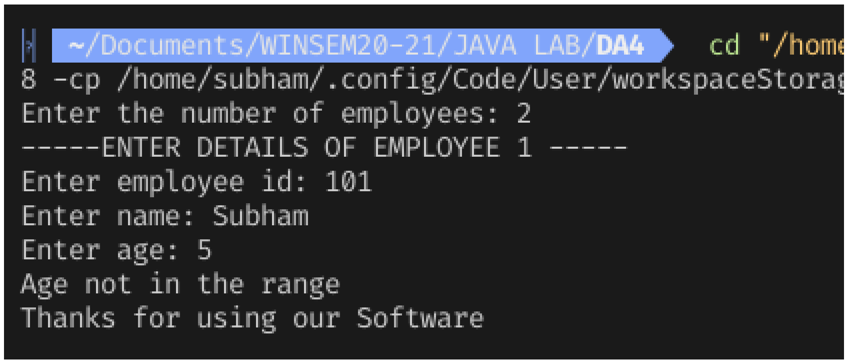
Question 7:

Code:

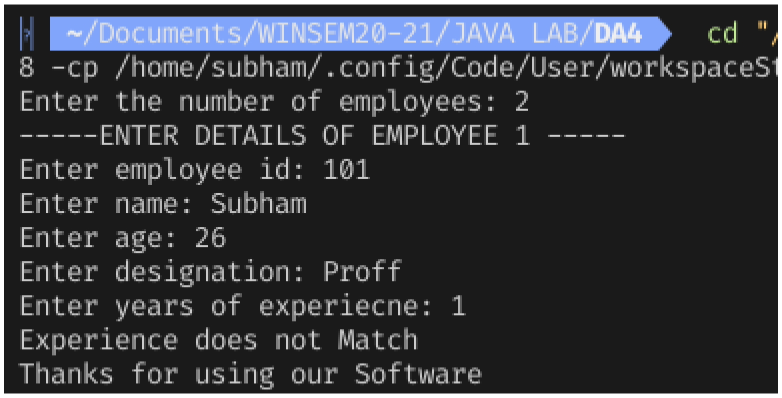
```java
import java.util.InputMismatchException;
import java.util.Scanner;

public class qs7 {
   public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       Scanner sc1 = new Scanner(System.in);
       try {
           System.out.print("Enter the number of employees: ");
           int n = sc.nextInt();
           employee emparr[] = new employee[n];
           for (int i = 0; i < n; i++) {
               System.out.println("-----ENTER DETAILS OF EMPLOYEE " + (i +
1) + " -----");
               System.out.print("Enter employee id: ");
               String empid = sc1.nextLine();
               System.out.print("Enter name: ");
               String name = sc1.nextLine();
               System.out.print("Enter age: ");
               int age = sc.nextInt();
               if (age > 60 || age < 25) {
                   throw new AgeException("Age not in the range");
               }
               System.out.print("Enter designation: ");
               String designation = sc1.nextLine();
               System.out.print("Enter years of experiecne: ");
               int yearsOfExperience = sc.nextInt();
               if(yearsOfExperience > 20 || yearsOfExperience < 5) {
                   throw new ExperienceException("Experience does not
Match");
               }
               System.out.print("Enter department: ");
               String department = sc1.nextLine();
               System.out.print("Enter salary: ");
               int salary = sc.nextInt();
               if (salary > 500000 || salary < 5000) {
                   throw new SalaryException("Salary does not fall within
the range");
               }
               emparr[i] = new employee(empid, name, age, designation,
yearsOfExperience, department, salary);

           }
       } catch (SalaryException e) {
           System.out.println(e.getMessage());
       } catch (AgeException e) {
           System.out.println(e.getMessage());
       } catch (ExperienceException e) {
           System.out.println(e.getMessage());
       } catch (InputMismatchException e) {
           System.out.println("Enter the correct type of data");
       } catch (Exception e) {
           System.out.println(e.getMessage());
       } finally {
           sc.close();
           sc1.close();
           System.out.println("Thanks for using our Software");
       }
   }
}

class employee {
   String empid;
   String name;
   int age;
   String designation;
   int yearsOfExperience;
   String department;
   int salary;

   employee(String empid, String name, int age, String designation, int
yearsOfExperience, String department,
           int salary) {
       this.empid = empid;
       this.name = name;
```

```
        this.age = age;
        this.designation = designation;
        this.yearsOfExperience = yearsOfExperience;
        this.department = department;
        this.salary = salary;
    }
}

class ExperienceException extends Exception {
    ExperienceException(String S) {
        super(S);
    }
}

class SalaryException extends Exception {
    SalaryException(String S) {
        super(S);
    }
}

class AgeException extends Exception {
    AgeException(String S) {
        super(S);
    }
}
```

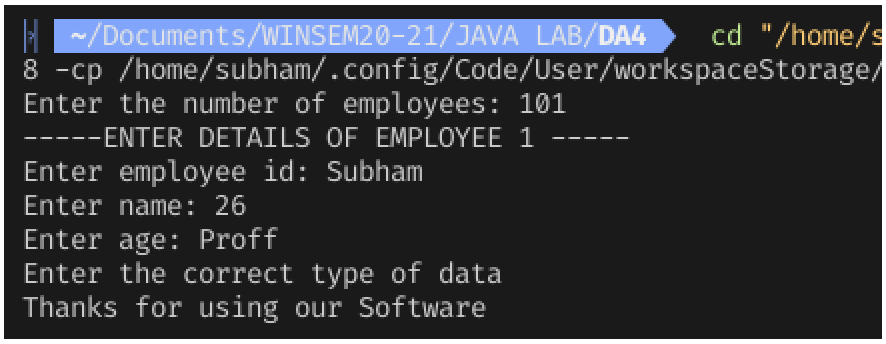
Output:



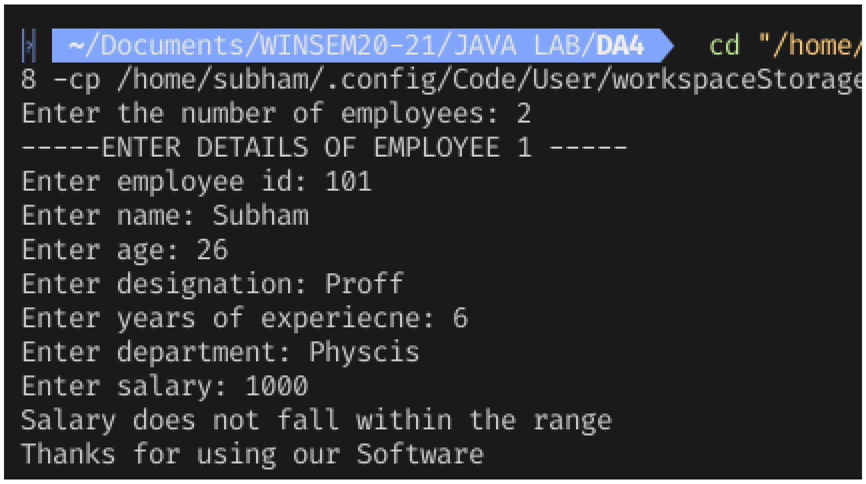
```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home
8 -cp /home/subham/.config/Code/User/workspaceStorag
Enter the number of employees: 2
-----ENTER DETAILS OF EMPLOYEE 1 -----
Enter employee id: 101
Enter name: Subham
Enter age: 5
Age not in the range
Thanks for using our Software
```



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/
8 -cp /home/subham/.config/Code/User/workspaceSt
Enter the number of employees: 2
-----ENTER DETAILS OF EMPLOYEE 1 -----
Enter employee id: 101
Enter name: Subham
Enter age: 26
Enter designation: Proff
Enter years of experiecne: 1
Experience does not Match
Thanks for using our Software
```



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/s
8 -cp /home/subham/.config/Code/User/workspaceStorage/
Enter the number of employees: 101
-----ENTER DETAILS OF EMPLOYEE 1 -----
Enter employee id: Subham
Enter name: 26
Enter age: Proff
Enter the correct type of data
Thanks for using our Software
```



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/
8 -cp /home/subham/.config/Code/User/workspaceStorage
Enter the number of employees: 2
-----ENTER DETAILS OF EMPLOYEE 1 -----
Enter employee id: 101
Enter name: Subham
Enter age: 26
Enter designation: Proff
Enter years of experiecne: 6
Enter department: Physcis
Enter salary: 1000
Salary does not fall within the range
Thanks for using our Software
```

Question 8:

Exceptions are bound to happen in the following part of the code:

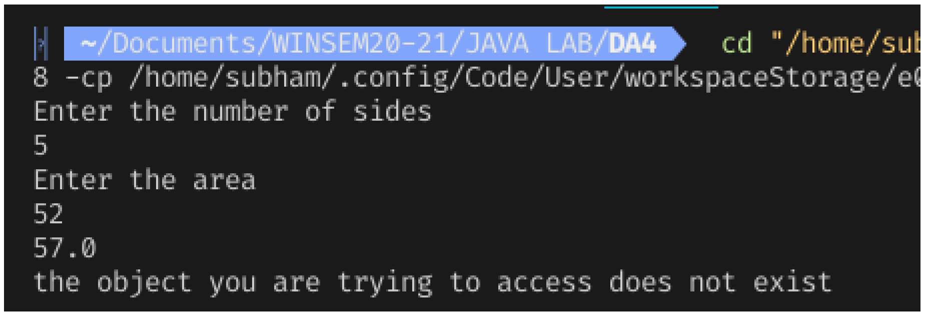
1. s[0] = new shape(); - so if s[0] gets dereferenced later then ***`NullPointerException`*** could occur when we reference s[0] in future
2. A NullPointerException will happen in the given code because shapename of the object is never initialized or assigned, hence while referring ***`this.shapename`***, it will invoke a ***`NullPointerException`***
3. In ***`setshapeDetails()`*** when the user is asked to enter number of sides of type integer and the area of type double, if the user enter any other character that does not correspond to that of type integer or double then ***`InputMismatchException`*** could occur.
4. The constructor of class ***`FileOutputStream`*** can throw exception ***`FileNotFoundException`***, hence that needs to be handled as well.
5. ***`write()`*** function of ***`FileOutputStream`*** can throw error ***`IOException`*** and hence that needs to handled as well

Corrected Code:

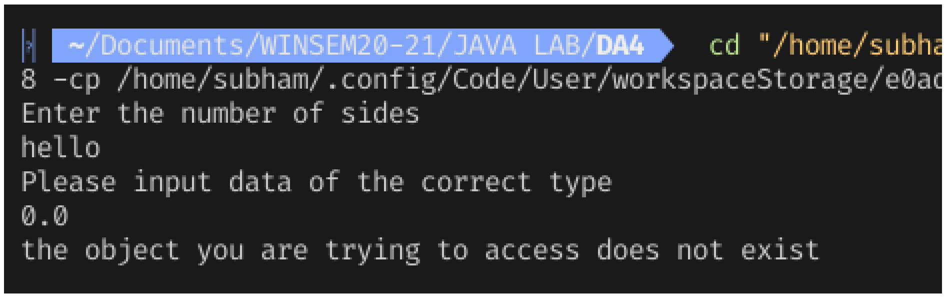
```
import java.io.*;
import java.util.InputMismatchException;
import java.util.Scanner;

public class qs8 {
   public static void main(String[] args) {
       try {
           shape s[] = new shape[4];
           s[0] = new shape();
           s[0].setShapeDetails();
           s[0].display_details();
           s[0].display_shapename();
           s[0].write_datatoFile();
       } catch (NullPointerException e) {
           System.out.println("the object you are trying to access does
not exist");
       } catch (Exception e) {
           System.out.println(e.getMessage());
       }
```

```
    }
}

class shape {
    int numsides;
    double area;
    String shapename;

    public void setShapeDetails() {
        try {
            System.out.println("Enter the number of sides");
            Scanner input = new Scanner(System.in);
            this.numsides = input.nextInt();//
            System.out.println("Enter the area");
            this.area = input.nextDouble();
        } catch (InputMismatchException e) {
            System.out.println("Please input data of the correct type");
        }

    }

    public void display_details() {
        System.out.println(numsides + area);
    }

    public void display_shapename() {
        if (this.shapename.equals("circle")) {
            System.out.println("its a circle");
        }
    }

    public void write_datatoFile() {
        try {
            File f = new File("satish.txt");
            FileOutputStream fout = new FileOutputStream(f);
            fout.write(numsides);
            fout.close();
        } catch (FileNotFoundException e) {
            System.out.println("The file was not found");
        } catch (IOException e) {
            System.out.println("There was some issue in writing the
contents to the file");
        }

    }
}
```

Output:



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/sub
8 -cp /home/subham/.config/Code/User/workspaceStorage/e
Enter the number of sides
5
Enter the area
52
57.0
the object you are trying to access does not exist
```



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subha
8 -cp /home/subham/.config/Code/User/workspaceStorage/e0ac
Enter the number of sides
hello
Please input data of the correct type
0.0
the object you are trying to access does not exist
```

Question 9:

Thread Interference and Memory Inconsistency:

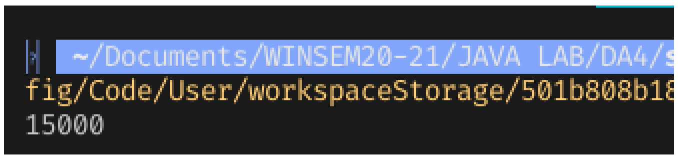
If two threads running try to access the same piece of data in memory then there is inconsistency if that piece of data is being updated in the memory. There could be inconsistent results in case the functions run by the two threads are not synchronized. This is known as Thread interference and Memory inconsistency. This can be overcome by running synchronized functions on the separate threads.

Code:

```
public class qs9 {
   public static void main(String[] args) throws InterruptedException {
       account satish = new account();
       account ramesh = new account();
       Thread trans1 = new Thread(new Runnable() {
           @Override
           public void run() {
               for (int i = 0; i < 500; i++) {
                   satish.withdraw(10);
               }
           }
       });
       Thread trans2 = new Thread(new Runnable() {
           @Override
           public void run() {
               for (int i = 0; i < 500; i++) {
                   ramesh.withdraw(10);
               }
           }
       });
       trans1.start();
       trans2.start();
       trans1.join();
       trans2.join();
       satish.getbalance();
   }
}

class account {
   private int balance;

   public account() {
       this.balance = 20000;
   }

   public synchronized void withdraw(int withdraw_amount) {
       this.balance = this.balance - withdraw_amount;
   }

   public void getbalance() {
       System.out.println(this.balance);
   }
}
```

The above code runs a synchronized functions on two threads and hence they are running their own critical sections. So the code will show no Thread interference or Memory inconsistency.

Output:



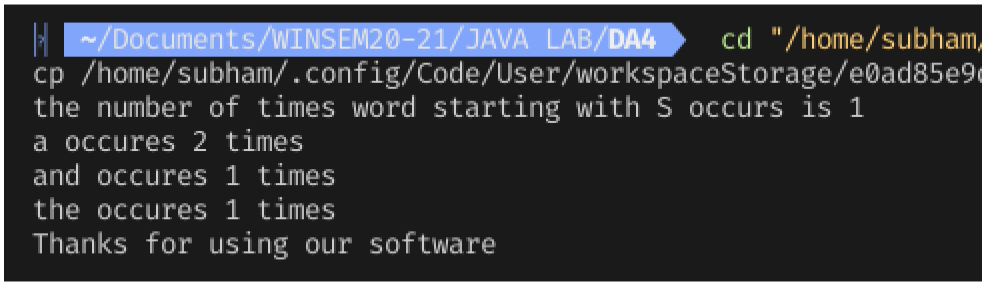
```
  ~/Documents/WINSEM20-21/JAVA LAB/DA4/s
fig/Code/User/workspaceStorage/501b808b18
15000
```

Question 10:

Code:

```
import java.util.*;
import java.io.*;

public class qs10 {
    public static void main(String[] args) throws Throwable {
        try {
            Thread thread1 = new Thread((Runnable) () -> {
                Scanner sc;
                try {
                    sc = new Scanner(new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/sample1.txt"));
                    int arr[] = { 0, 0, 0 };
                    while (sc.hasNext()) {
                        String word = sc.next();
                        if (word.compareTo("a") == 0) {
                            arr[0]++;
                        } else if (word.compareTo("and") == 0) {
                            arr[1]++;
                        } else if (word.compareTo("the") == 0) {
                            arr[2]++;
                        }
                    }
                    System.out.println("a occures " + arr[0] + " times");
                    System.out.println("and occures " + arr[1] + " times");
                    System.out.println("the occures " + arr[2] + " times");
                    sc.close();
                } catch (FileNotFoundException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            });
            Thread thread2 = new Thread((Runnable) () -> {
                Scanner sc;
                try {
                    sc = new Scanner(new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/sample2.txt"));
                    int c = 0;
                    while (sc.hasNext()) {
                        String word = sc.next();
                        if (word.compareTo("a") != 0 &&
word.compareTo("and") != 0 && word.compareTo("the") != 0
                                && word.startsWith("S")) {
                            c++;
                        }
                    }
                    System.out.println("the number of times word starting
with S occurs is " + c);
                    sc.close();
                } catch (FileNotFoundException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }

            });
            Thread thread3 = new Thread((Runnable) () -> {
                System.out.println("Thanks for using our software");
            });
            thread1.start();
            thread1.setPriority(2);
            thread2.start();
            thread2.setPriority(2);
            thread1.join();
            thread2.join();
            thread3.start();
            thread3.setPriority(1);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Output:



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4   cd "/home/subham
cp /home/subham/.config/Code/User/workspaceStorage/e0ad85e9
the number of times word starting with S occurs is 1
a occures 2 times
and occures 1 times
the occures 1 times
Thanks for using our software
```

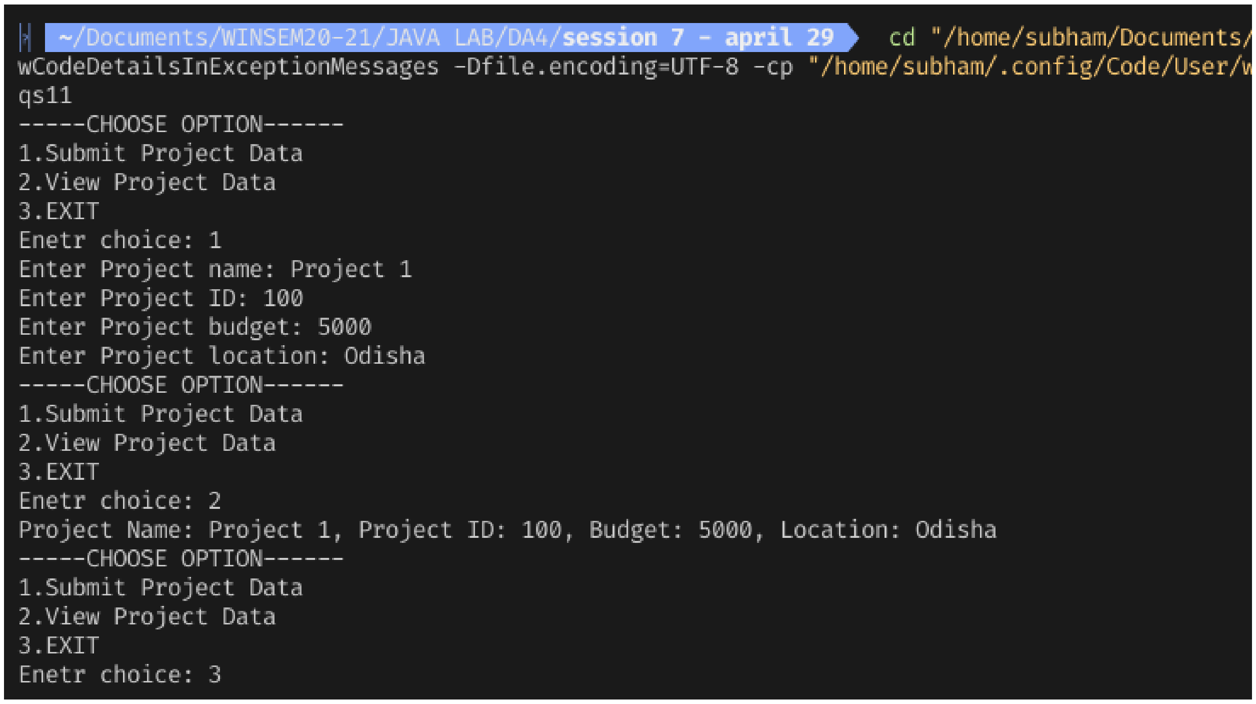
Question 11:

Code:

```java
import java.util.*;
import java.io.*;

public class qs11 {
   public static void main(String[] args) throws IOException,
ClassNotFoundException {
        Scanner sc = new Scanner(System.in);
        Scanner sc1 = new Scanner(System.in);
        while (true) {
            System.out.println("-----CHOOSE OPTION------");
            System.out.println("1.Submit Project Data");
            System.out.println("2.View Project Data");
            System.out.println("3.EXIT");
            System.out.print("Enetr choice: ");
            int n = sc.nextInt();
            if (n == 1) {
                System.out.print("Enter Project name: ");
                String ProjectName = sc1.nextLine();
                System.out.print("Enter Project ID: ");
                int projectID = sc.nextInt();
                System.out.print("Enter Project budget: ");
                int budget = sc.nextInt();
                System.out.print("Enter Project location: ");
                String location = sc1.nextLine();
                Project p = new Project(ProjectName, projectID, budget,
location);
                File obj = new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/project.txt");
                FileOutputStream fout = new FileOutputStream(obj);
                ObjectOutputStream objout = new ObjectOutputStream(fout);
                objout.writeObject(p);
                objout.close();
            } else if (n == 2) {
                File obj = new
File("/home/subham/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april
29/project.txt");
                FileInputStream fin = new FileInputStream(obj);
                ObjectInputStream objin = new ObjectInputStream(fin);
                Project parr[] = new Project[Project.total];
                for (int i = 0; i < Project.total; i++) {
                    parr[i] = (Project) objin.readObject();
                    parr[i].displayInfo();
                }
                objin.close();
            } else if (n == 3) {
                sc.close();
                sc1.close();
                System.exit(1);
            } else {
                System.out.println("INVALID CHOICE");
            }
        }
    }
}

class Project implements Serializable {
    String ProjectName;
    int projectID;
    int budget;
    String location;

    static int total = 0;

    Project(String ProjectName, int projectID, int budget, String location)
{
        this.ProjectName = ProjectName;
        this.projectID = projectID;
        this.budget = budget;
        this.location = location;
        total++;
    }
```

```java
    public void displayInfo() {
        System.out.println("Project Name: " + ProjectName + ", Project ID: 
" + projectID + ", Budget: " + budget
                + ", Location: " + location);
    }
}
```

Output:



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april 29   cd "/home/subham/Documents/
wCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -cp "/home/subham/.config/Code/User/w
qs11
-----CHOOSE OPTION------
1.Submit Project Data
2.View Project Data
3.EXIT
Enetr choice: 1
Enter Project name: Project 1
Enter Project ID: 100
Enter Project budget: 5000
Enter Project location: Odisha
-----CHOOSE OPTION------
1.Submit Project Data
2.View Project Data
3.EXIT
Enetr choice: 2
Project Name: Project 1, Project ID: 100, Budget: 5000, Location: Odisha
-----CHOOSE OPTION------
1.Submit Project Data
2.View Project Data
3.EXIT
Enetr choice: 3
```

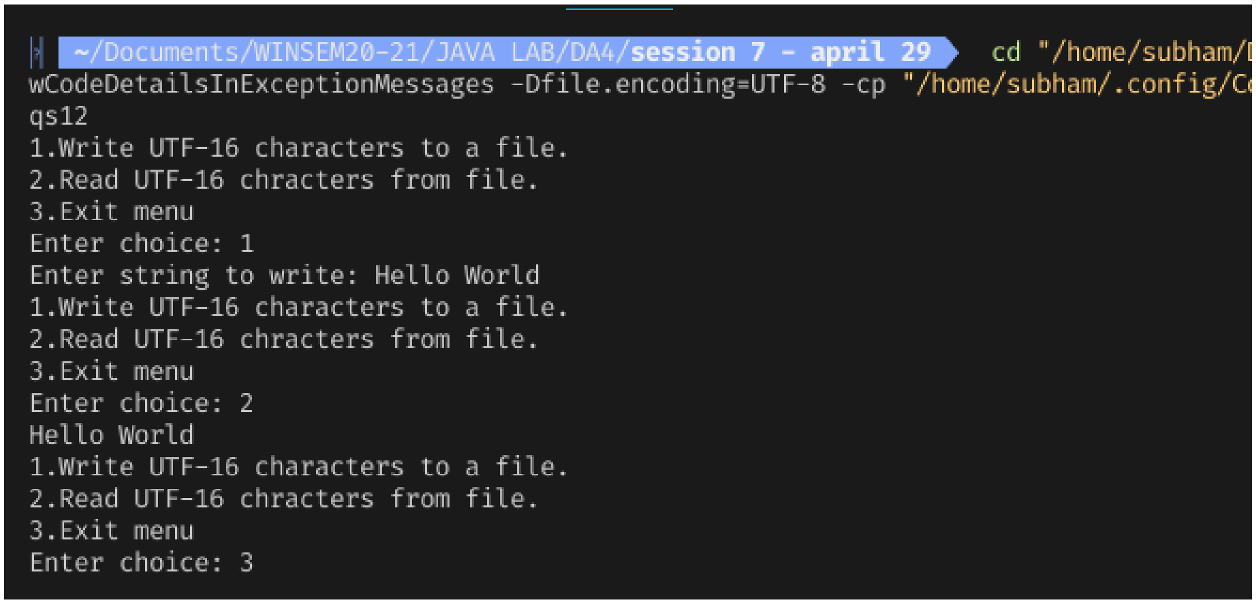
Question 12:

Code:

```java
import java.util.*;
import java.io.*;

public class qs12 {
   public static void main(String[] args) throws Throwable {
       try {
           File f = new File("/home/subham/Documents/WINSEM20-21/JAVA
LAB/DA4/session 7 - april 29/sample.txt");
           while (true) {
               System.out.println("1.Write UTF-16 characters to a file.");
               System.out.println("2.Read UTF-16 chracters from file.");
               System.out.println("3.Exit menu");
               Scanner sc1 = new Scanner(System.in);
               Scanner sc = new Scanner(System.in);
               System.out.print("Enter choice: ");
               int n = sc.nextInt();
               if (n == 1) {
                   OutputStreamWriter owrite = new OutputStreamWriter(new
FileOutputStream(f), "UTF16");
                   System.out.print("Enter string to write: ");
                   String s = sc1.nextLine();
                   owrite.write(s);
                   owrite.close();
               } else if (n == 2) {
                   InputStreamReader iread = new InputStreamReader(new
FileInputStream(f), "UTF16");
                   char c[] = new char[100];
                   iread.read(c);
                   for (char x : c) {
                       System.out.print(x);
                   }
                   System.out.println();
                   iread.close();
               } else if (n == 3) {
                   sc.close();
                   sc1.close();
```

```
                    System.exit(1);
                } else {
                    System.out.println("INVALID CHOICE");
                }
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Output:



```
 ~/Documents/WINSEM20-21/JAVA LAB/DA4/session 7 - april 29   cd "/home/subham/D
wCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -cp "/home/subham/.config/Co
qs12
1.Write UTF-16 characters to a file.
2.Read UTF-16 chracters from file.
3.Exit menu
Enter choice: 1
Enter string to write: Hello World
1.Write UTF-16 characters to a file.
2.Read UTF-16 chracters from file.
3.Exit menu
Enter choice: 2
Hello World
1.Write UTF-16 characters to a file.
2.Read UTF-16 chracters from file.
3.Exit menu
Enter choice: 3
```