Advanced OOPs Lab Report

Name: Soumalya Kundu

Roll No: 001810501033

BCSE UG 2

Soumalya Kundu
001810501033

Assignment 1

Introduction to JAVA

Q1. Write a program to accept two short integers from user and display the sum.

Source Code:

```
import java.util.Scanner;

class s1q1
{
   public static void main(String args[])
   {
      Scanner ip = new Scanner(System.in);
      short a,b;
      System.out.println("Enter 1st short number: ");
      a = ip.nextShort();
      System.out.println("Enter 2nd short number: ");
      b = ip.nextShort();
      int c = a+b;
      System.out.println("Answer: "+c);
   }
}
```

Output:

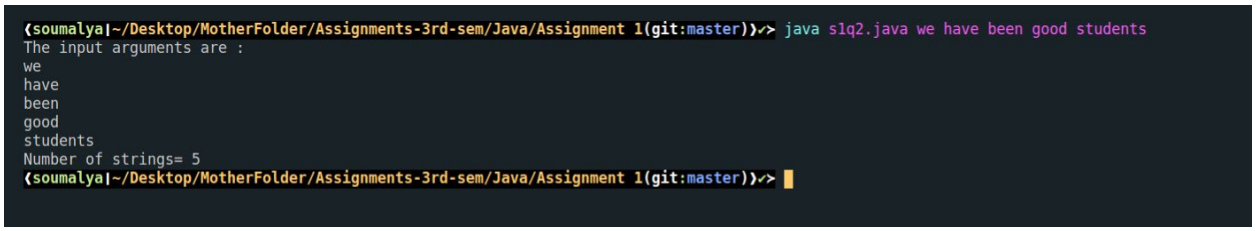
```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓> java s1q1.java
Enter 1st short number:
3
Enter 2nd short number:
4
Answer: 7
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓>
```

Q2: Write a program that accepts number of command line parameters and displays the parameters and count of such parameters.

Source Code:

```
import java.util.*;

class Prog1_2
{
	public static void main(String args[])
	{
		int p=0;
		System.out.println("The input arguments are :");
		for(p=0;p<args.length;p++)
		{
			System.out.println(args[p]);
		}
		System.out.println("Number of strings= "+p);
	}
}
```

Output:

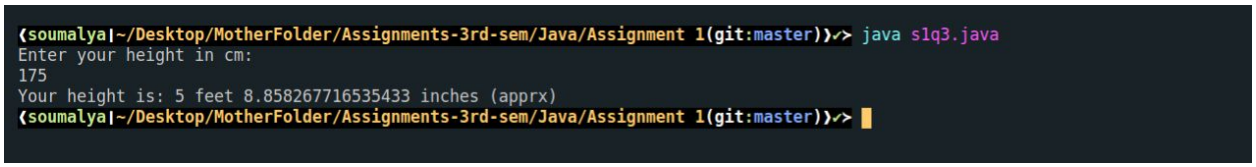


Q3. Write a program that accepts height in cm as int and displays the height in feet and inches. Assume, 1 inch equals to 2.54 cm and 1 foot equals to 30.5 cm.

Source Code:

```
import java.util.Scanner;
class s1q3
{
    public static void main(String args[])
    {
        System.out.println("Enter your height in cm: ");
        Scanner ip = new Scanner(System.in);
        int cm = ip.nextInt();
        int feet = (int)(cm/30.5);
        double d = cm - (30.5)*feet;
        d = d/2.54;
        System.out.println("Your height is: "+ feet +" feet " + d +" inches
(apprx)");
    }
}
```

Output:

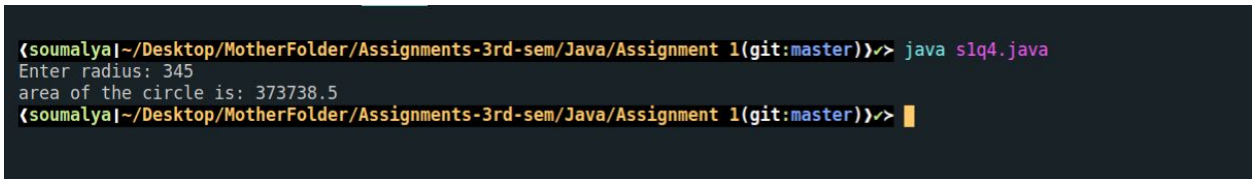


Q4. Write a program that accepts radius of a circle and displays area of the circle. Declare a constant pi equals to 3.14.

Source Code:

```
import java.util.Scanner;

class s1q4
{
   public static void main(String args[])
   {
      final static double pi = 3.14;
      System.out.print("Enter radius: ");
      int radius;
      Scanner ip = new Scanner(System.in);
      radius = ip.nextInt();
      double area = pi*radius*radius;
      System.out.println("area of the circle is: " + area);
   }
}
```

Output:



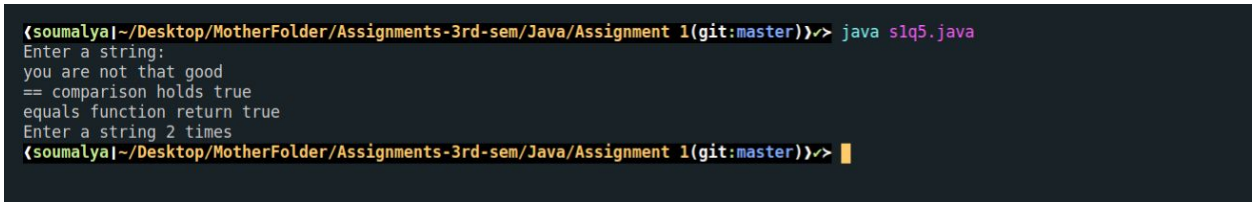
Q5. Write a program that accepts a String and assigns it to another. Check the outcome of comparison with == and equals() method. Take two Strings and put same input for them. Repeat the equality checking. Observe the outcome.

Source Code:

```
import java.util.*;

class Prog1_5
{
	public static void main(String args[])
	{
		String s;
		Scanner sc= new Scanner(System.in);
		System.out.println("Enter a string: ");
		s= sc.next();
		String str;
		str= s;
		if(s==str)
			System.out.println("== comparison holds true");
		if(s.equals(str))
			System.out.println("equals function return true");
		System.out.println("Enter a string 2 times");
		s=sc.next();
```

```
            str=sc.next();
            if(s==str)
                System.out.println("== comparison holds true");
            if(s.equals(str))
                System.out.println("equals function return true");
        }
}
```

Output:



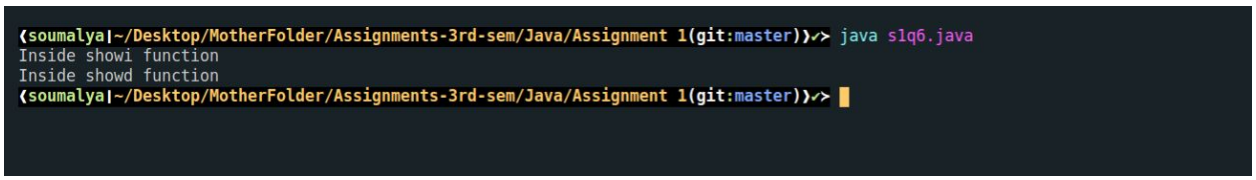
Q6. Write a program where class contains void show(int) to display the argument passed. Call the function once with short as actual parameter and again double as actual parameter. Add another function as void show(double) . Repeat the calls. Observe the outcomes in each case.

Source Code:

```
import java.util.*;

class Prog1_6
{
        static void show(int i)
        {
                System.out.println("Inside show function");
        }
```

```
    static void show(double d)
    {
        System.out.println("Inside show function");
    }

    public static void main(String args[])
    {
        short s=5;
        show(s);
        double d= 5.56;
        show(d);
    }
}
```

Output:



Q7. Design and implement Student class with roll, name and score as attributes. It will have methods to set attributes (attribute values passed as arguments), display the attributes, copy (that copies the content of invoking object to another object passed as argument). Verify that methods are working properly.

Source Code:

```
import java.util.*;

class Student
{
	int roll;
	String name;
	double score;

	public void set(int r, String n, double s)
	{
		roll=r;
		name=n;
		score=s;
	}

	public void show()
	{
		System.out.println("Roll: "+roll);
		System.out.println("Name: "+name);
		System.out.println("Score: "+score);
	}

	public void copy(Student s)
	{
		roll= s.roll;
```

```
            name= s.name;
            score= s.score;
        }
}

class Prog1_7
{
        public static void main(String args[])
        {
            Student s1= new Student();
            Student s2= new Student();
            Scanner sc= new Scanner(System.in);
            System.out.println("Enter student details (roll,name,score): ");
            int i=sc.nextInt();
            String s=sc.next();
            double d= sc.nextDouble();
            s1.set(i,s,d);
            System.out.println("Student 1: ");
            s1.show();
            s2.copy(s1);
            System.out.println("Student 2: ");
            s2.show();
        }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))↵> javac s1q7.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))↵> java s1q7
Enter student details (roll,name,score):
45 soumalya 234
Student 1:
Roll: 45
Name: soumalya
Score: 234.0
Student 2:
Roll: 45
Name: soumalya
Score: 234.0
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))↵>
```

Q8. Add constructors in the Student class of earlier problem so that objects can be created with i) roll only, ii) roll and name only, iii) roll, name and score, iv) no value. Also include a copy constructor. Check whether constructors are working or not. Verify, copy constructor results into deep coy or not.

Source Code:

```
import java.util.Scanner;

class Student
{
	int roll;
	String name;
	double score;

	public Student()
	{
		roll=0;
	}

	public Student(int r)
	{
		roll=r;
	}

	public Student(int r, String n)
	{
		roll=r;
		name=n;
	}

	public Student(int r, String n, double s)
```

```
	{
		roll=r;
		name=n;
		score=s;
	}

	public Student(Student s)
	{
		roll=s.roll;
		name=s.name;
		score=s.score;
	}

	public void set(int r, String n, double s)
	{
		roll=r;
		name=n;
		score=s;
	}

	public void show()
	{
		System.out.println("Roll: "+roll);
		System.out.println("Name: "+name);
		System.out.println("Score: "+score);
	}

}

class Prog1_8
{
	public static void main(String args[])
	{
		Scanner sc= new Scanner(System.in);
		System.out.println("Enter student details (roll,name,score): ");
```

```
        int i=sc.nextInt();
        String s=sc.next();
        double d= sc.nextDouble();
        Student s1= new Student();
        Student s2= new Student(i);
        Student s3= new Student(i,s);
        Student s4= new Student(i,s,d);
        Student s5= new Student(s4);
        System.out.println("Student 1: ");
        s1.show();
        System.out.println("Student 2: ");
        s2.show();
        System.out.println("Student 3: ");
        s3.show();
        System.out.println("Student 4: ");
        s4.show();
        System.out.println("Student 5: ");
        s5.show();
        s4.set(i+1, s, d);
        if(s4.roll==s5.roll)
            System.out.println("Shallow copy");
        else
            System.out.println("Deep copy");
    }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓> javac s1q8.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓> java Prog1_8
Enter student details (roll,name,score):
45 soumalya 231
Student 1:
Roll: 0
Name: null
Score: 0.0
Student 2:
Roll: 45
Name: null
Score: 0.0
Student 3:
Roll: 45
Name: soumalya
Score: 0.0
Student 4:
Roll: 45
Name: soumalya
Score: 231.0
Student 5:
Roll: 45
Name: soumalya
Score: 231.0
Deep copy
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓>
```

Q9. Design a BankAcct class with account number, balance and interest rate as attribute. Interest rate is same for all account. Support must be there to initialize, change and display the interest rate. Also supports are to be there to return balance and calculate interest.

Source Code:

```
import java.util.Scanner;

class BankAcc
{
   String accNo;
   long balance;
   double rate = 4.50;

   public BankAcc(String accNo, long balance, double rate )
   {
      this.accNo = accNo;
      this.balance = balance;
```

```
        this.rate = rate;
    }

    void displayRate()
    {
        System.out.println("Interest rate is: " + rate + "% :)");
    }

    void setRate(double rate )
    {
        this.rate = rate;
    }

    void calculateInterest()
    {
        double time;
        System.out.print("Enter time(in year): ");
        Scanner ip = new Scanner(System.in);
        time = ip.nextDouble();
        double money = (balance*time*rate)/100;
        System.out.println("Interest is: " + money);
    }
}

public class s1q9
{
    public static void main(String args[])
    {
        BankAcc b;
        b = new BankAcc("abc123456",100000,6.25);
        b.calculateInterest();
        b.setRate(9.25);
        b.calculateInterest();

    }
```

```
}
```

Output:

Q10. Design a Metric class that supports Kilometre to Mile conversion with distance in Kilometre as argument and Mile to Kilometre conversion with distance in mile as argument. Assume, one Mile equals 1.5 Kilometre.

Source Code:

```
import java.util.Scanner;

class Metric
{
   void convertInKM(double mile)
   {
      double km = mile*1.5;
      System.out.println("distance in kilometers: " + km);
   }
   void convertInMile(double km)
   {
      double mile = km/1.5;
      System.out.println("distance in miles: " + mile);
```

```
    }

}

public class s1q10
{
    public static void main(String args[])
    {
        Metric m = new Metric();
        Scanner ip = new Scanner(System.in);
        double dist;
        System.out.print("Enter your distance in km: ");
        dist = ip.nextDouble();
        m.convertInMile(dist);

        System.out.print("Enter your distance in mile: ");
        dist = ip.nextDouble();
        m.convertInKM(dist);
        ip.close();
    }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓> javac s1q10.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓> java s1q10
Enter your distance in km: 45
distance in miles: 30.0
Enter your distance in mile: 23
distance in kilometers: 34.5
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 1(git:master))✓>
```

Q11. Each Instructor has name and phone number. One can view instructor information and set the information. Textbook has a title, author name and publisher. One can set the data for a textbook and view the same. Each course has a course name, instructor and text book. One can set the course data and view the same. Design and implement the classes .

Source Code:

```
import java.util.*;

class Instructor{
	String name;
	int phno;

	void view(){
		System.out.println("Name: "+ name + "\nContact: "+ phno);
	}
	void set(){
		Scanner sc = new Scanner(System.in);
		name = sc.next();
		System.out.println("Enter phone number : ");
		phno = sc.nextInt();
	}
}
class Textbook{
	String title;
	String name;
	String publisher;
	void view(){
		System.out.println("Title: "+ title + "\nName: " + name +
"\nPublisher: " + publisher);
	}
```

```
        void set(){
                Scanner sc = new Scanner(System.in);
                title = sc.next();
                name = sc.next();
                publisher = sc.next();
        }
}

class Course{
        Instructor ins;
        Textbook t;
        Course(){
                ins = new Instructor();
                t = new Textbook();
        }
        void view(){
                t.view();
                ins.view();
        }
        void set(){
                t.set();
                ins.set();
        }
        public static void main(String args[]){
                Course c = new Course();
                c.set();
                c.view();
        }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))✓> javac s2q1.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))✓> java s2q1
Enter Customer name: Soumalya
Enter Customer ID: 6789
Enter Customer phone number: 8765749839
Enter your required loan amount: 5464
Loan credited!!! Enjoy your day!!!
Enter Customer name: Soumalya
Enter Customer ID: 6789
Enter Customer phone number: 839304840475
Enter your required loan amount: 54637
Loan credited!!! Enjoy your day!!!
Enter Customer name: ^C
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))✗>
```

Soumalya Kundu
001810501033

Assignment 2

Introduction to JAVA

Q1. Bank ID and customer related Question.

Source Code:

```
import java.lang.reflect.Constructor;
import java.util.Scanner;

class Customer
{
   String customerID;
   String name;
   String phoneNumber;
   static final Long creditLimit = 6000000;
   Long totalLoan;

   public Customer()
   {
      totalLoan = 0;
   }

   void setDetails()
   {
      System.out.print("Enter Customer name: ");
      Scanner ip = new Scanner(System.in);
      this.name = ip.nextLine();
      System.out.print("Enter Customer ID: ");
      this.customerID = ip.nextLine();
      System.out.print("Enter Customer phone number: ");
      this.phoneNumber = ip.nextLine();
```

```
    }

    void showDetails()
    {
        System.out.println("Customer name: " + name);
        System.out.println("Customer ID: " + customerID);
        System.out.println("Customer phone number: " + phoneNumber);
        System.out.println("Credit Limit: "+ creditLimit);
        System.out.println("Total loan can be credited: "
+(creditLimit-totalLoan));

    }

    void creditLoan()
    {
        int loan;
        System.out.print("Enter your required loan amount: ");
        Scanner ip = new Scanner(System.in);
        loan = ip.nextInt();
        if(totalLoan+loan >= creditLimit)
        {
            System.out.println("CreditLimit reached!! Loan cannot be
credited!!!");
        }
        else
        {
            totalLoan = totalLoan + loan;
            System.out.println("Loan credited!!! Enjoy your day!!!");
    }

}

public class s2q1
{
    public static void main (String args[])
```

```
    {
        Customer[] c;
        c = new Customer[10];
        int i;
        for(i=0;i<10;i++)
        {
            c[i] = new Customer();
            c[i].setDetails();
            c[i].creditLoan();
        }

    }
}
```

Output:

Q2. Faculty and Student related question.

Source Code:

```
import java.util.Scanner;


class Address
{
    String premises_no;
    String street;
    String city;
```

```
String pin;
String state;

Address()
{
    premises_no = "";
    street = "";
    city = "";
    pin = "";
    state = "";
}

public void getDetails()
{
    System.out.println("premises no: " + premises_no);
    System.out.println("Street no: " + street);
    System.out.println("city: " + city);
    System.out.println("pin no: " + pin);
    System.out.println("state: " + state);
}

public void setPremisesNo()
{
    System.out.print("Enter Premises no: ");
    Scanner ip = new Scanner(System.in);
    this.premises_no = ip.nextLine();
}

public void setStreetNo()
{
    System.out.print("Enter Street No: ");
    Scanner ip = new Scanner(System.in);
    this.street = ip.nextLine();
}
```

```
    public void setCity()
    {
        System.out.print("Enter City: ");
        Scanner ip = new Scanner(System.in);
        this.city = ip.nextLine();
    }

    public void setState()
    {
        System.out.print("Enter State: ");
        Scanner ip = new Scanner(System.in);
        this.state = ip.nextLine();
    }

    public void setPin()
    {
        System.out.print("Enter Pin: ");
        Scanner ip = new Scanner(System.in);
        this.pin = ip.nextLine();
    }

    public void setDetails()
    {
        this.setPremisesNo();
        this.setStreetNo();
        this.setPin();
        this.setCity();
        this.setState();
    }

}
class Person
{
    String name;
    Address address;
```

```
String phone;
String email;

Person()
{
    address = new Address();
}
public void setDetails()
{
    this.setName();
    this.setPhone();
    this.setAddress();
    this.setEmail();
}

public void setName()
{
    System.out.print("Enter name: ");
    Scanner ip = new Scanner(System.in);
    this.name = ip.nextLine();
}

public void setPhone()
{
    System.out.print("Enter Phone number: ");
    Scanner ip = new Scanner(System.in);
    this.phone = ip.nextLine();
}

public void setEmail()
{
    System.out.print("Enter Email ID: ");
    Scanner ip = new Scanner(System.in);
    this.email = ip.nextLine();
}
```

```
    public void setAddress()
    {
        this.address.setDetails();
    }

    public void getDetails()
    {
        //System.out.println("#####################################");
        System.out.println("\nName: " + name);
        System.out.println("Phone: " + phone);
        System.out.println("Email ID: " + email);
        //System.out.println("Address:------");
        address.getDetails();
        //System.out.println("#####################################");
    }
}

class Student extends Person
{
    String roll;
    String course;

    Student()
    {
        super();
        roll = "";
        course = "";
    }

    public void setDetails()
    {
        super.setDetails();
        System.out.println("Enter your roll no: ");
        Scanner ip = new Scanner(System.in);
```

```
        roll = ip.nextLine();
        System.out.println("Enter your course of study: ");
        course = ip.nextLine();
    }

    public void getDetails()
    {
        super.getDetails();
        System.out.println("Roll No: " + this.roll);
        System.out.println("Course Of Study: " + this.course);
    }
}

class Faculty extends Person
{
    String id;
    String specialization;

    Faculty()
    {
        super();
        this.id = "";
        this.specialization = "";
    }

    public void setDetails()
    {
        super.setDetails();
        System.out.print("Enter your Employee ID: ");
        Scanner ip = new Scanner(System.in);
        id = ip.nextLine();
        System.out.print("Enter your specialization : ");
        specialization = ip.nextLine();
    }
```

```
    public void getDetails()
    {
        super.getDetails();
        System.out.println("Emplyee ID: " + this.id);
        System.out.println("Subject of Specialization: " + this.specialization);
    }
}


public class Prog2_2
{
    public static void main(String args[])
    {
        Faculty f = new Faculty();
        Student s = new Student();

        System.out.println("Enter faculty details: ");
        f.setDetails();
        System.out.println("#####################################");
        f.getDetails();
        System.out.println("#####################################");

        System.out.println("Enter Stident details: ");
        s.setDetails();
        System.out.println("#####################################");
        s.getDetails();
        System.out.println("#####################################");
    }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))↵> javac s2q2.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))↵> java s2q2
Enter faculty details:
Enter name: Souvik
Enter Phone number: 6578494
Enter Premises no: 345
Enter Street No: 18/3 MG road
Enter Pin: 700 073
Enter City: Kolkata
Enter State: west bengal
Enter Email ID: souvik@gmail.com
Enter your Employee ID: 5679303
Enter your specialization : Math
################################

Name: Souvik
Phone: 6578494
Email ID: souvik@gmail.com
premises no: 345
Street no: 18/3 MG road
city: Kolkata
pin no: 700 073
state: west bengal
Emplyee ID: 5679303
Subject of Specialization: Math
################################
Enter Stident details:
Enter name: Soumalya
Enter Phone number: 736203704238
Enter Premises no: Ronalak
Enter Street No: Rasbihari Avn.
Enter Pin: 700 012
Enter City: Kolkata
Enter State: west bengal
Enter Email ID: kdjf'f@gmail.com
Enter your roll no:
23
Enter your course of study:
23
################################

Name: Soumalya
```

Q3. The famous Library design question.

Source Code:

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

//import javax.imageio.plugins.tiff.ExifTIFFTagSet;

class Book {
    int bookid;
    String title;

    Book() {
```

```
    }

    Book(int id, String t) {
        bookid = id;
        title = t;
    }

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Book ID : ");
        bookid = sc.nextInt();
        sc.nextLine();
        System.out.print("Title : ");
        title = sc.nextLine();
        //sc.close();

    }

    void printBook() {
        System.out.println("Book ID : " + bookid + "\nTitle : " + title);
    }
}

class BookList {
    int size;
    int max_size;
    int bookids[];
    String titles[];
    int total_copies[];
    int available[];

    BookList(int s) {
        max_size = s;
        size = 0;
```

```
        bookids = new int[max_size];
        titles = new String[max_size];
        total_copies = new int[max_size];
        available = new int[max_size];
    }

    int existsID(int id) {
        for (int i = 0; i < size; i++) {
            if (id == bookids[i])
                return i;
        }
        return -1;
    }

    int addBook(Book b) {
        int val = existsID(b.bookid);
        if (val == -1) {
            System.out.println("Unique Book ID identified");
            if (size < max_size) {
                bookids[size] = b.bookid;
                titles[size] = b.title;
                total_copies[size] = 1;
                available[size] = 1;
                size++;
                return 1;
            } else {
                System.out.println("Cannot add anymore books");
                return -1;
            }
        } else {
            System.out.println("Book ID is not unique, book cannot be added to
List");
            return -1;
        }
    }
```

```
    int addCopies(int id, int new_val) {
        int val = existsID(id);
        if (val == -1) {
            System.out.println("Book does not exist");
            return val;
        } else {
            available[val] += new_val;
            total_copies[val] += new_val;
            return 1;
        }
    }

    int addCopy(int id){
         int val = existsID(id);
         if(val != -1)
         {
                 available[val]++;
                 return 1;
         }
         return -1;
    }

    int removeCopies(int id, int sub) {
        int val = existsID(id);
        if (val == -1) {
            System.out.println("Book does not exist");
            return val;
        } else {
            available[val] -= sub;
            if (available[val] < 0) {
                System.out.println("No more books left, cannot be
issued/removed");
                return -1;
            }
```

```
            return 1;
        }

    }

    void displayBookInformation(int id) {
        int val = existsID(id);
        if (val != -1) {
            Book b = new Book(bookids[val], titles[val]);
            b.printBook();
        } else {
            System.out.print("Book does not Exist.\n");
        }
    }

    void displayList() {
        System.out.println("Book ID\tTitle\tCopies Left");
        for (int i = 0; i < size; i++) {
            System.out.println(bookids[i] + "\t" + titles[i] + "\t" + available[i]);
        }
    }

}

class Date {
    int day, month, year;
    Date(int a, int b, int c){
        day = a;
        month = b;
        year = c;
    }

    void print() {
        System.out.println(day + "/" + month + "/" + year);
    }
```

```
}

class Member {
    int memberid;
    String name;
    Date dob;
    int books;
    static int total_books = 5;

    Member() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Member ID : ");
        memberid = sc.nextInt();
        sc.nextLine();
        System.out.println("Name : ");
        name = sc.nextLine();
        books = 0;
        dob = new Date(1, 1, 2000);
        //sc.close();
    }

    Member(int id, String t, Date dob) {
        memberid = id;
        name = t;
        this.dob = dob;
        books = 0;
    }

    void printMember() {
        System.out.println("Member ID : " + memberid + "\nName : " + name +
"\nBooks Issued : " + books);
        dob.print();
    }
```

```
}

class MemberList {
    int size;
    int max_size;
    Member members[];

    MemberList(int s) {
        max_size = s;
        s = 0;
        members = new Member[max_size];

    }

    int addBooks(int id) {
        int val = existsMember(id);
        if (members[val].books < Member.total_books) {
            members[val].books++;
            return 1;
        } else {
            System.out.println("Maximum books issued");
            return -1;
        }
    }

    int removeBooks(int id) {
        int val = existsMember(id);
        members[val].books--;
        return 1;
    }

    int existsMember(int x) {
        for (int i = 0; i < size; i++) {
            if (members[i].memberid == x) {
                return i;
```

```
            }

        }
        return -1;
    }

    int insertMember(Member m) {
        int val = existsMember(m.memberid);
        if (val == -1) {
            if (size < max_size) {
                members[size] = m;
                size++;
                return 1;
            } else {
                System.out.println("Member List is full");
                return -1;
            }
        } else {
            System.out.println("Member already exists");
            return -1;
        }
    }

    void displayList() {
        for (int i = 0; i < size; i++) {
            members[i].printMember();
        }
    }

    void displayMemberInformation(int val) {
        int id = existsMember(val);
        System.out.println(id);
        members[id].printMember();
    }
}
```

```
class Transaction {
    Member m;
    Book b;
}

class TransactionList {
    int bookids[];
    int memberids[];
    int max_size;
    int size;

    TransactionList(int s) {
        max_size = s;
        size = 0;
        memberids = new int[s];
        bookids = new int[s];
    }

    int addTransaction(int memid, int bookid) {
        if (size < max_size) {
            memberids[size] = memid;
            bookids[size] = bookid;
            size++;
            return 1;
        } else {
            System.out.println("Too many Transactions");
            return -1;
        }
    }

    int existsTransaction(int memid, int bookid) {
        for (int i = 0; i < size; i++) {
            if (memberids[i] == memid && bookids[i] == bookid) {
                return i;
```

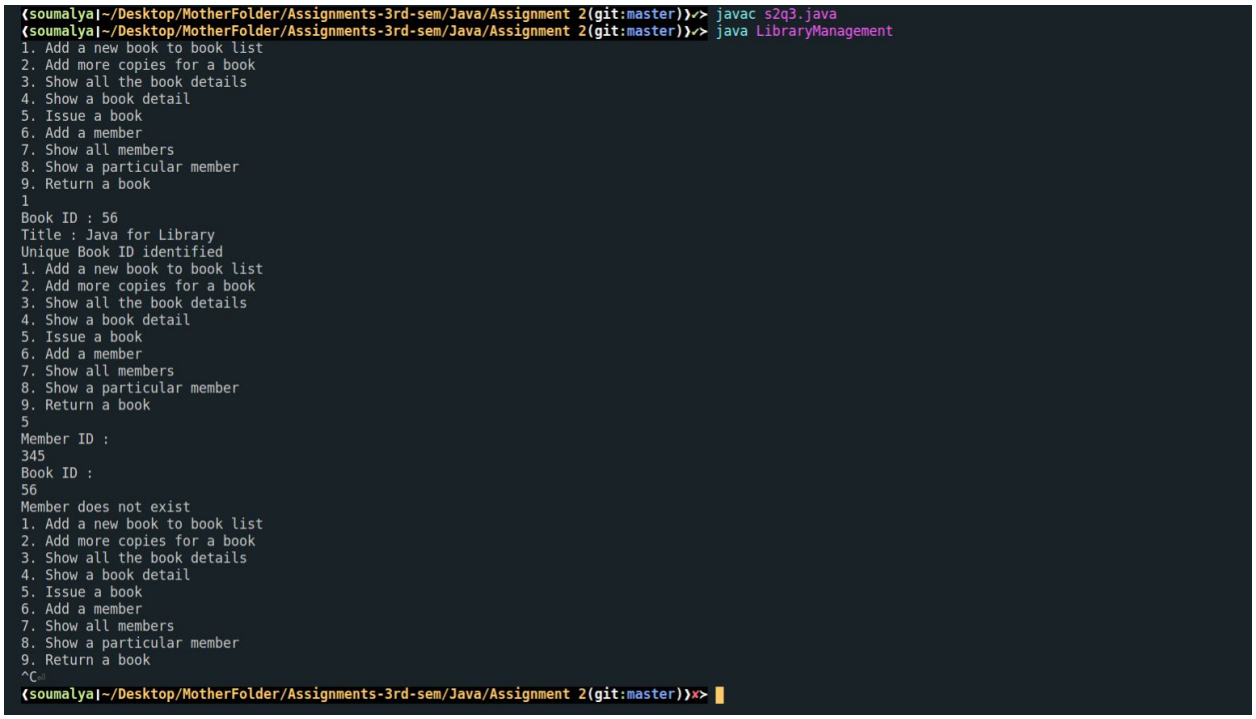
```
            }
        }
        return -1;
    }

    int removeTransaction(int memid, int bookid) {
        int val = existsTransaction(memid, bookid);
        if (val == -1) {
            System.out.println("No such transaction\n");
            return -1;
        } else {
            memberids[val] = -9999;
            return 1;

        }
    }

}

class LibraryManagement {
    MemberList ml;
    BookList bl;
    TransactionList tl;

    LibraryManagement() {
        ml = new MemberList(10000);
        bl = new BookList(10000);
        tl = new TransactionList(10000);
    }

    void run() throws IOException{
        int choice = 10;
        Scanner sc = new Scanner(System.in);
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
```

```
        do {
            System.out.println(
                    "1. Add a new book to book list\n2. Add more copies for a
book\n3. Show all the book details\n4. Show a book detail\n5. Issue a
book\n6. Add a member\n7. Show all members\n8. Show a particular
member\n9. Return a book");
            choice = Integer.parseInt(br.readLine());
            switch (choice) {
            case 1:
                Book b = new Book();
                b.input();
                bl.addBook(b);
                break;

            case 2:
                System.out.println("Book ID : ");
                int id = sc.nextInt();
                int final_val;
                System.out.println("How many more copies were added : ");
                final_val = sc.nextInt();
                bl.addCopies(id, final_val);
                break;
            case 3:
                bl.displayList();
                break;
            case 4:
                System.out.println("Book ID : ");
                int id1 = sc.nextInt();
                bl.displayBookInformation(id1);
                break;
            case 5:
                System.out.println("Member ID : ");
                int memid = sc.nextInt();
                System.out.println("Book ID : ");
                int id2 = sc.nextInt();
```

```
    if (ml.existsMember(memid) != -1) {
      {
            if(ml.addBooks(memid) != -1){
            if(bl.existsID(id2) != -1){
          if(bl.removeCopies(id2, 1)!= -1)
            tl.addTransaction(memid, id2);
          }
          else{
            System.out.println("Book does not exist\n");

          }
       }
            }
    } else {
       System.out.println("Member does not exist");
    }
    break;
case 6:
    Member m = new Member();
    ml.insertMember(m);
    break;
case 7:
    ml.displayList();
    break;
case 8:
    System.out.println("Member ID : ");
    int mem = sc.nextInt();
    ml.displayMemberInformation(mem);
    break;
case 9:
    System.out.println("Member ID : ");
    int m1 = sc.nextInt();
    System.out.println("Book ID : ");
    int b1 = sc.nextInt();
    if (tl.removeTransaction(m1, b1) != -1) {
```

```
                ml.removeBooks(m1);
                bl.addCopy(b1);
            }
            break;
        default:
            System.out.println("Invalid Input\n");

        }


    } while (choice != -1);
    //sc.close();
  }

  public static void main(String args[]) throws IOException{
    LibraryManagement lm = new LibraryManagement();
    lm.run();
  }
}
```

Output:



Q4. Interface for Library question.

Source Code:

```
import java.util.Scanner;
import java.lang.Exception;

class MyException extends Exception{
    MyException(String msg){
        super(msg);
    }
}

class Excep{
    private String roll,name;
```

```
    private float score =0.0f;
    public void setScore(float score){
        this.score = score;
    }
    public void setName(String name){
        this.name = name;
    }
    public void setRoll(String roll){
        this.roll = roll;
    }
    public String getName(){
        return name;
    }
    public String getRoll(){
        return roll;
    }
    public static void main(String []args){
        Excep e = new Excep();
        Scanner sc = new Scanner(System.in);
        try{
            e.setName("Rahul");
            e.setRoll("12345678");
            System.out.println("Name : "+e.getName());
            System.out.println("Roll : "+e.getRoll());
            System.out.println("Enter the score ");
            e.score = Float.parseFloat(sc.nextLine());
            if(e.score<0 || e.score>100){
                throw new MyException("####@@@@Not in Range
Exception.Please Enter a value betwwn 0 & 100####@@@@");
            }
            e.setScore(e.score);
            System.out.println("Score "+e.score);
        }
        catch(MyException ep){
            System.out.println("Caught the exception ");
```

```
            System.out.println(ep.getMessage());
        }
        finally{
            System.out.println("Finally Executed !!!");
        }
    }
}
```

Output:

Q5. student class to deal with exceptions.

Source Code:

```
import java.util.Scanner;
import java.lang.Exception;

class MyException extends Exception{
    MyException(String msg){
        super(msg);
    }
}

class Excep{
    private String roll,name;
    private float score =0.0f;
    public void setScore(float score){
        this.score = score;
    }
    public void setName(String name){
        this.name = name;
    }
    public void setRoll(String roll){
        this.roll = roll;
```

```
    }
    public String getName(){
        return name;
    }
    public String getRoll(){
        return roll;
    }
    public static void main(String []args){
        Excep e = new Excep();
        Scanner sc = new Scanner(System.in);
        try{
            e.setName("Rahul");
            e.setRoll("12345678");
            System.out.println("Name : "+e.getName());
            System.out.println("Roll : "+e.getRoll());
            System.out.println("Enter the score ");
            e.score = Float.parseFloat(sc.nextLine());
            if(e.score<0 || e.score>100){
                throw new MyException("####@@@@Not in Range
Exception.Please Enter a value betwwn 0 & 100####@@@@");
            }
            e.setScore(e.score);
            System.out.println("Score "+e.score);
        }
        catch(MyException ep){
            System.out.println("Caught the exception ");
            System.out.println(ep.getMessage());
        }
        finally{
            System.out.println("Finally Executed !!!");
        }
    }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))✓> javac s2q5.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))✓> java Excep
Name : Soumalya
Roll : 001810501033
Enter the score
234
Caught the exception
--------Not in Range Exception.Please Enter a value betwwn 0 & 100--------
Finally Executed !!!
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 2(git:master))✓>
```

Q6. Introduction to the wrapper classes.

Source Code:

```
import java.util.Scanner;

public class s2q6
{
	public static void main(String args[])
	{
		//basic type to wrapper
		int a = 5;
		Integer iobj = new Integer(a);

		//object to basic type
		Integer iobj2 = new Integer(4);
		int p;
		p = iobj2.intValue();

		//basic to String type
		int a2 = 10;
		String s = Integer.toString(a2);

		//string to numeric obj
		String mama = new String("1122334");
		int a3 = Integer.parseInt(mama);
```

```
            //string to wrapper class
            Integer iobj1 = Integer.valueOf("5678");
        }
}
```

Output:

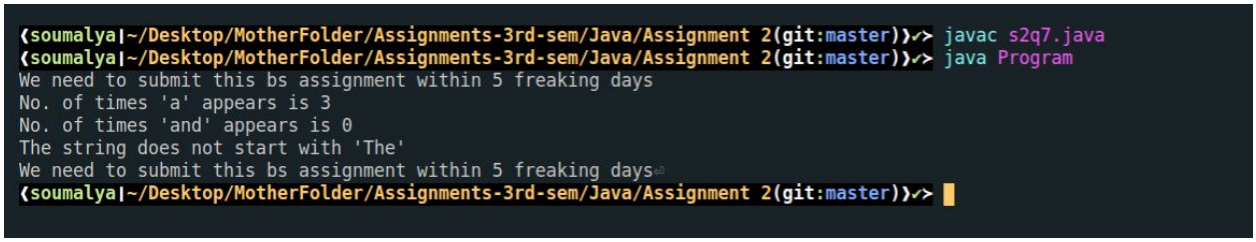
Q7. String Manipulation question.

Source Code:

```
import java.util.Scanner;

class Program{
    public static void main(String[]args){
        Scanner sc=new Scanner(System.in);
        String str = sc.nextLine();
            int cnt1=0,cnt2=0;
            int i;
            for(i=0;i<str.length();i++)
                if(str.charAt(i)=='a')
                    cnt1++;
            System.out.println("No. of times 'a' appears is "+cnt1);
            for(i=0;i<str.length()-3;i++)
                if(str.substring(i,i+3).equals("and"))
                    cnt2++;
            System.out.println("No. of times 'and' appears is "+cnt2);
            if(str.startsWith("The"))
                System.out.println("The string starts with 'The'");
            else
                System.out.println("The string does not start with 'The'");
            char[] ch=str.toCharArray();
            for(i=0;i<ch.length;i++)
                System.out.print(ch[i]);
    }
```

```
}
```

Output:



Assignment 3
Introduction to JAVA

Q1. Employee in a department question.

Source Code:

```
import java.util.*;

class Department{
    String code;
    String location;
    String name;
```

```
public Department() {}

public Department(String code, String location, String name) {
    this.code = code;
    this.location = location;
    this.name = name;
}

public String getCode() {
    return this.code;
}

public void setCode(String code) {
    this.code = code;
}

public String getLocation() {
    return this.location;
}

public void setLocation(String location) {
    this.location = location;
}

public String getName() {
    return this.name;
}

public void setName(String name) {
    this.name = name;
}

public Department code(String code) {
    this.code = code;
    return this;
```

```
    }

    public Department location(String location) {
        this.location = location;
        return this;
    }

    public Department name(String name) {
        this.name = name;
        return this;
    }

    @Override
    public boolean equals(Object o) {
        if (o == this)
            return true;
        if (!(o instanceof Department)) {
            return false;
        }
        Department department = (Department) o;
        return Objects.equals(code, department.code) &&
Objects.equals(location, department.location) && Objects.equals(name,
department.name);
    }


    @Override
    public String toString() {
        return "DeptCode = " + getCode() + "\nLocation = " + getLocation() +
"\nName = " + getName() ;

    }

    public void getData(){
        Scanner sc1 = new Scanner(System.in);
```

```
        Scanner sc2 = new Scanner(System.in);
        Scanner sc3 = new Scanner(System.in);
        System.out.print("Enter department code: ");
        String code = sc1.nextLine();
        System.out.print("Enter department location: ");
        String location = sc2.nextLine();
        System.out.print("Enter department name: ");
        String name = sc3.nextLine();
        setName(name);
        setCode(code);
        setLocation(location);
    }


    void display(){
        System.out.println(this);
    }


}


class Employee{
    String code;
    String name;
    int basic;
    String dept_code;

    public Employee() {
    }

    public Employee(String code, String name, int basic, String dept_code) {
        this.code = code;
        this.name = name;
        this.basic = basic;
```

```
    this.dept_code = dept_code;
}

public String getCode() {
    return this.code;
}

public void setCode(String code) {
    this.code = code;
}

public String getName() {
    return this.name;
}

public void setName(String name) {
    this.name = name;
}

public int getBasic() {
    return this.basic;
}

public void setBasic(int basic) {
    this.basic = basic;
}

public String getDept_code() {
    return this.dept_code;
}

public void setDept_code(String dept_code) {
    this.dept_code = dept_code;
}
```

```
    public Employee code(String code) {
        this.code = code;
        return this;
    }

    public Employee name(String name) {
        this.name = name;
        return this;
    }

    public Employee basic(int basic) {
        this.basic = basic;
        return this;
    }

    public Employee dept_code(String dept_code) {
        this.dept_code = dept_code;
        return this;
    }

    @Override
    public boolean equals(Object o) {
        if (o == this)
            return true;
        if (!(o instanceof Employee)) {
            return false;
        }
        Employee employee = (Employee) o;
        return Objects.equals(code, employee.code) &&
Objects.equals(name, employee.name) && basic == employee.basic &&
Objects.equals(dept_code, employee.dept_code);
    }

    @Override
    public String toString() {
```

```
        return "Code = " + getCode() + "\nName = " + getName() + "\nBasic
pay = " + getBasic() + "\nDept code = " + getDept_code();
    }

    public void getData(){
        Scanner sc1 = new Scanner(System.in);
        Scanner sc2 = new Scanner(System.in);
        Scanner sc3 = new Scanner(System.in);
        Scanner sc4 = new Scanner(System.in);
        System.out.print("Enter employee code\n");
        String code = sc1.nextLine();
        System.out.print("Enter employee name\n");
        String name = sc2.nextLine();
        System.out.print("Enter basic pay\n");
        int basic = sc3.nextInt();
        System.out.print("Enter department code\n");
        String dept_code = sc4.nextLine();
        setName(name);
        setBasic(basic);
        setCode(code);
        setDept_code(dept_code);
    }

    public void show()
    {
        System.out.println(this);
    }

}

class Dept_List{
    ArrayList<Department> arr;
    public Dept_List() {
        arr = new ArrayList<Department>();
    }
```

```
void add(){
    Department d = new Department();
    for ( int i = 0 ; i < arr.size() ; i++ ){
        if ( arr.get(i).getCode().equals(d.getCode())){
            System.out.println("Not unique department!! Try agian!!");
        }
    }
    d.getData();
    arr.add(d);
}

void show( String code){
    for ( int  i = 0 ; i < arr.size() ; i ++ ){
        if ( arr.get(i).getCode().equals(code)){
            arr.get(i).display();
        }
    }
    System.out.println("Not found");
}

void calTotalPay(Emp_List e , String code){
    int f =0;
    for ( int i =0 ; i < arr.size() ; i ++ ){
        if ( arr.get(i).getCode().equals(code)){
            f = 1;
            break;
        }
    }
    if ( f == 0){
        System.out.println("wrong code");
        return ;       }
    int sum = 0;
    for ( int i = 0 ; i < e.arr.size() ; i++ ){
        if ( e.arr.get(i).getDept_code().equals(code)){
            sum += e.arr.get(i).getBasic();
```

```
            }
        }
        System.out.println("Total pay: "+ sum);
    }
}

class Emp_List{
    ArrayList<Employee> arr;
    public Emp_List(){
        arr = new ArrayList<Employee>();
    }

    void add(){
        Employee d = new Employee();
        for (int i = 0; i < arr.size(); i++) {
            if (arr.get(i).getCode() == d.getCode()) {
                System.out.println("Employee exists!!!");
            }
        }
        d.getData();
        arr.add(d);
    }

    void display( String code){
        for ( int i = 0; i < arr.size() ; i++){
            if ( arr.get(i).getCode().equals(code) ){
                arr.get(i).show();
                Dept_List d = new Dept_List();
                d.show(arr.get(i).getDept_code());
                return ;
            }
        }
    }
    void displayAll(){
        for ( int i =0 ; i  < arr.size() ; i ++ ){
```

```
            System.out.println(arr.get(i));
        }
    }

    void remove( String code){
        for ( int i =0 ; i < arr.size() ; i++){
            if ( arr.get(i).getCode().equals(code)){
                arr.remove(i);
                return;
            }
        }
        System.out.println("Employee Not present!! Terminating!!");
    }

    void modify( String code , int new_basic){
        for ( int i =0 ; i < arr.size() ; i++){
            if ( arr.get(i).getCode().equals(code)){
                arr.get(i).setBasic(new_basic);
                return;
            }
        }
        System.out.println("Employee Not found!! Try again!!");
    }
}



class s3p1{
     public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        Emp_List e = new Emp_List();
        Dept_List d = new Dept_List();
        while(true){
            System.out.println("....................................................................");
            System.out.println("1 add a new Employee");
```

```
System.out.println("2 Add a new department");
System.out.println("3 Display all the employee details");
System.out.println("4 Find total basic pay for a department");
System.out.println("5 Remove an employee");
System.out.println("6.Modify basic pay of an employee");
System.out.println("7.exit");
System.out.println(".....");
//System.out.println("6 Sort employees");
int choice = sc.nextInt();
switch(choice)
{
   case 1:
      e.add();
      break;
   case 2:
      d.add();
      break;
   case 3:
      e.displayAll();
      break;
   case 4:
   {
      System.out.print("Enter code\n");
      Scanner sc1 = new Scanner(System.in);

      String code = sc1.nextLine();

      d.calTotalPay(e, code);
      break;
   }
   case 5:
   {
      System.out.print("Enter employee code\n");
      Scanner sc2 = new Scanner(System.in);
      String code = sc2.nextLine();
```

```
                e.remove(code);
                break;
            }
            case 6:
            {
                System.out.print("Enter employee code\n");
                Scanner sc3 = new Scanner(System.in);
                String code = sc3.nextLine();
                Scanner sc4 = new Scanner(System.in);
                int new_basic = sc4.nextInt();

                e.modify(code,new_basic);
                break;
            }
            case 7:
            {
                break;
            }
            default:
                System.out.println("Wrong entry!! try again!!");
                break;
  }
      }

    }
}
```

#Part2 of Question 1:

```
import java.util.*;

class Dept {
   String deptCode;
```

```
    String deptName;
    String location;

    public Dept() {}

    public Dept(String deptCode , String deptName , String location) {
        this.deptCode = deptCode;
        this.deptName = deptName;
        this.location = location;
    }

    public void getData() {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter DeptCode : ");
        deptCode = in.nextLine();
        System.out.print("Enter DeptName : ");
        deptName = in.nextLine();
        System.out.print("Enter Location : ");
        location = in.nextLine();
        System.out.print("\n");
    }
}

class Emp {
    String empCode;
    String empName;
    float basic;
    String deptCode;

    public Emp() {}

    public Emp(String empCode, String empName, float basic, String
deptCode) {
        this.empCode = empCode;
        this.empName = empName;
```

```
        this.basic = basic;
        this.deptCode = deptCode;
    }

    public void getData() {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter EmpCode : ");
        empCode = in.nextLine();
        System.out.print("Enter EmpName : ");
        empName = in.nextLine();
        System.out.print("Enter Basic Pay : ");
        basic = Float.parseFloat(in.nextLine());
        System.out.print("Enter DeptCode : ");
        deptCode = in.nextLine();
        System.out.print("\n");
    }

    public void updateBasic(float basic) {
        this.basic = basic;
    }

    public String toString() {
        return "\nEmpCode: " + empCode + "\nEmpName: " + empName +
"\nBasicPay: " + basic + "\nDeptCode: " + deptCode;
    }
}

class DeptList {
    LinkedList<Dept> list;

    public DeptList() {
        list = new LinkedList<>();
    }

    public boolean isPresent(String deptCode) {
```

```
        for(Dept dept : list) {
            if(dept.deptCode.equals(deptCode)) {
                return true;
            }
        }
        return false;
    }

    public void add(Dept obj) {
        if(!isPresent(obj.deptCode)) {
            list.add(obj);
        }
    }
}

class EmpCodeCmp implements Comparator<Emp> {
    public int compare(Emp a, Emp b) {
        return a.empCode.compareTo(b.empCode);
    }
}

class DeptCodeCmp implements Comparator<Emp> {
    public int compare(Emp a, Emp b) {
        return a.deptCode.compareTo(b.deptCode);
    }
}

class BasicCmp implements Comparator<Emp> {
    public int compare(Emp a, Emp b) {
        if(a.basic > b.basic) return -1;
        else if(a.basic < b.basic) return 1;
        else return 0;
    }
}
```

```
class EmpList {
    LinkedList<Emp> list;

    public EmpList() {
        list = new LinkedList<>();
    }

    public boolean isUsed(String empCode) {
        for(Emp emp : list) {
            if(emp.empCode.equals(empCode)) {
                return true;
            }
        }
        return false;
    }

    public void add(Emp obj , DeptList dlist) {
        if(!isUsed(obj.empCode) && dlist.isPresent(obj.deptCode)) {
            list.add(obj);
        }
    }

    public void printEmp(String empCode) {
        if(!isUsed(empCode)) {
            System.out.println("Employee not found!!\n");
        }
        for(Emp emp : list) {
            if(emp.empCode.equals(empCode)) {
                System.out.println(emp);
                break;
            }
        }
    }

    public void printAll() {
```

```
    for(Emp emp : list) {
        System.out.println(emp);
    }
}

public float totalBasic(String deptCode) {
    float res = 0;
    for(Emp emp : list) {
        if(emp.deptCode.equals(deptCode)) {
            res = res + emp.basic;
        }
    }
    return res;
}

public void remove(String empCode) {
    if(!isUsed(empCode)) return;

    int index = -1;
    for(int i = 0; i < list.size(); i++) {
        if(list.get(i).empCode.equals(empCode)) {
            index = i;
            break;
        }
    }

    if(index != -1) list.remove(index);
}

public void modifyBasic(String empCode , float basic) {
    if(!isUsed(empCode)) return;

    for(int i = 0; i < list.size(); i++) {
        Emp obj = list.get(i);
        if(obj.empCode.equals(empCode)) {
```

```
                obj.updateBasic(basic);
                list.add(i, obj);
                list.remove(i+1);
                break;
            }
        }
    }

    public void sortByBasic() {
        Collections.sort(list, new BasicCmp());
    }

    public void sortByEmpCode() {
        Collections.sort(list, new EmpCodeCmp());
    }

    public void sortByDeptCode() {
        Collections.sort(list, new DeptCodeCmp());
    }
}

public class Prog3_1b {
    public static void main(String[] args) {
        Dept d1 = new Dept();
        d1.getData();

        Emp e1  = new Emp(), e2 = new Emp();
        e1.getData();
        e2.getData();

        DeptList dlist = new DeptList();
        dlist.add(d1);
        EmpList elist = new EmpList();
        elist.add(e1 , dlist);
        elist.add(e2 , dlist);
```

```
        elist.sortByBasic();
        elist.printAll();
    }
}
```

Output:

Q2. Bank Account and Balance Question.

Source Code:

```
import java.util.*;

class Account{
        private       int accno;
        private String name;
        private double balance;
        Account(int code)
        {
          accno=code;
        }
        Account(Account A)
      {
        accno=A.retaccno();
        balance=A.retbalance();
        name=A.retname();
      }
        Account(int accno,String name,double balance)
        {
          this.accno=accno;
          this.name=name;
          this.balance=balance;
```

```
        }
        public int retaccno()
        {
          return accno;
        }
        public double retbalance()
        {
          return balance;
        }
    public String retname()
    {
        return name;
        }

        public void display()
        {
          System.out.println(this);
        }
        public String toString()
        {
        return("\nAccount number: "+accno+"\nName: "+name+" \nBalance:
"+balance+"\n");
        }
        public boolean equals(Account t)
        {
          if(t.accno==accno)
           return true;
        else
        return false;
        }
}
public class s3p2{
        public static void main(String args[])
        {
                HashMap<Integer,Account> h=new HashMap();
```

```
 int code;
 double balance;
 String name=new String();
 Scanner S=new Scanner(System.in);
 Scanner S1=new Scanner(System.in);
 while(true)
 {
System.out.println("Press 1 to create the Account list");
System.out.println("Press 2 to display details of an account");
System.out.println("Press 3 to display all accounts");
System.out.println("Enter your choice");
int choice=S.nextInt();
         switch(choice)
         {
        case 1:
        System.out.println("Enter the name");
        name=S1.nextLine();
                System.out.println("Enter the code");
                code=S.nextInt();
        System.out.println("Enter the balance ammount");
                balance=S.nextDouble();
                Account A=new Account(code,name,balance);
        h.put(code,A);
        break;
                case 2:
                System.out.println("Enter the code");
                code=S.nextInt();

                if(h.containsKey(code))
            {
                                Account Acc=new Account(h.get(code));
                                Acc.display();
                }
            else{
               System.out.println("The account does not exist");
```

```
                    }
                break;
                case 3:
                Set <Map.Entry<Integer,Account>> s;
                s = h.entrySet();
                for(Map.Entry m:s){
                        System.out.println(m.getValue());
                }
                break;
                default:
                System.exit(1);
                  }
            }
        }
}
```

Output:

Q3. Filename and Directory List question.

Source Code:

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

public class Prog3_3 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String filename;

        System.out.println("Enter any Filename : ");
        filename = in.nextLine();
```

```
        File f = new File(filename);
        if(f.exists()) {
            if(f.isDirectory()) {
                File[] list = f.listFiles();

                for(int i = 0; i < list.length; i++) {
                    System.out.println(list[i].getName());
                }
            }
            else {
                if(f.canRead()) {
                    System.out.println("File is readable!!");
                }
                else {
                    System.out.println("File is not readable!!");
                }

                if(f.canWrite()) {
                    System.out.println("File is writable!!");
                }
                else {
                    System.out.println("File is not writable!!");
                }
            }
        }
        else {
            System.out.println("File doesnot exist!!");
        }
    }
}
```

Output:

Q4. Text file and basic file handling with java question.

Source Code:

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

class FileCreate
{
   public static void main(String args[]) throws IOException
   {
      //FileWriter fw = new FileWriter();
      File obj = new File("Random.txt");
      try
      {
         if(obj.createNewFile())
         {
            System.out.println("File created: " + obj.getName());
         }
         else
         {
            System.out.println("File already exists...!!! no problem!!");
         }
      }
      catch(IOException e)
      {
         System.out.println("An error occured!!");
         e.printStackTrace();
      }
      try
      {
         FileWriter fobj = new FileWriter("Random.txt",true);
         System.out.println("Enter the number of strings: ");
         int a;
         Scanner ip = new Scanner(System.in);
```

```
    a = ip.nextInt();
    for(int i = 0; i <a;i++)
    {
        System.out.println("Enter a name: ");
        Scanner ip1 = new Scanner(System.in);
        String s = ip1.nextLine();
        fobj.write(s+"\n");
    }
    fobj.close();
    System.out.println("File write successfully\n");
}

catch(Exception e)
{
    System.out.println("Error occured while writing!!");
    e.printStackTrace();
}

try
{
    Scanner readObj = new Scanner(obj);
    System.out.println("The names you entered were: ");
    while(readObj.hasNextLine())
    Scanner ip = new Scanner(System.in);
    a = ip.nextInt();
    {
        String data = readObj.nextLine();
        System.out.println(data);
    }
    readObj.close();
    System.out.println("File read and closed successfully!!");
}
catch(Exception e)
{
    System.out.println("Error occured while reading!!");
```

```
            e.printStackTrace();
        }

    }
}
```

Output:

Q5. Student class and file reader question.

Source Code:

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

class Student implements Serializable {
    int roll;
    String name;
    float score;

    public Student() {}

    public Student(int roll , String name , float score) {
        this.roll = roll;
        this.name = name;
        this.score = score;
    }

    public void getData() {
        Scanner in = new Scanner(System.in);

        System.out.print("Enter Name: ");
        name = in.nextLine();
        System.out.print("Enter Roll: ");
        roll = Integer.parseInt(in.nextLine());
        System.out.print("Enter Score: ");
```

```
        score = Float.parseFloat(in.nextLine());
    }

    @Override
    public String toString() {
        return "\nName: " + name + "\nRoll: " + roll + "\nScore: " + score;
    }
}

public class Prog3_5 {
    public static void main(String[] args) {
        try {
            String filename;
            Scanner in = new Scanner(System.in);

            System.out.print("Enter filename: ");
            filename = in.nextLine();

            FileOutputStream f = new FileOutputStream(new File(filename));
            ObjectOutputStream o = new ObjectOutputStream(f);

            String choice;
            do {
                Student obj = new Student();
                obj.getData();
                o.writeObject(obj);
                System.out.println("\nWant to write more entries: (yes/no)? ");
                choice = in.nextLine();
            }while(choice.equals("yes"));
            o.close();
            f.close();

            FileInputStream fi = new FileInputStream(new File(filename));
            ObjectInputStream oi = new ObjectInputStream(fi);
```

```
            Student obj = new Student();
            System.out.println("File Contents : ");
            while(true) {
                try {
                    obj = null;
                    obj = (Student)oi.readObject();
                    if(obj != null) {
                        System.out.println(obj);
                    }
                }
                catch(EOFException e) {
                    oi.close();
                    fi.close();
                    break;
                }
            }
        }
        catch(Exception e) {
            System.out.println("Error!!");
            e.printStackTrace();
        }
    }
}
```

Output:

Soumalya Kundu
001810501033

Assignment 4
Introduction to JAVA

Q1. Thread application in java.

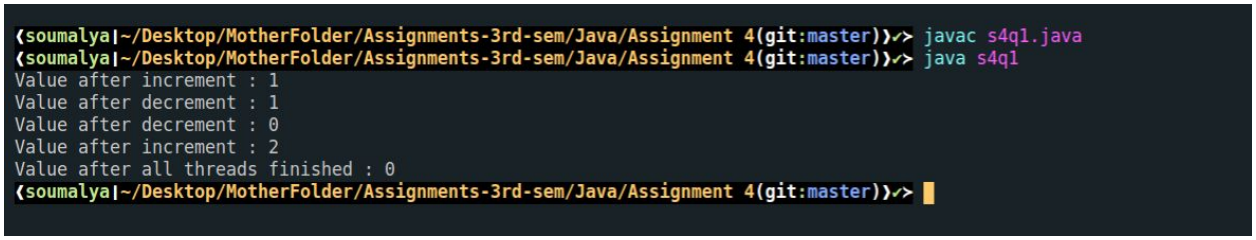
Source Code:

```
class Var {
    int n;

    public Var() {
        n = 0;
    }

    public void inc(int delta) {
        n += delta;
        System.out.println("Value after increment : " + n);
    }

    public void dec(int delta) {
        n -= delta;
        System.out.println("Value after decrement : " + n);
    }
}

class IncThread implements Runnable{
    Var v;
    Thread t;
```

```
    public IncThread(Var v , String threadName) {
        this.v = v;
        t = new Thread(this , threadName);
    }

    public void run() {
        v.inc(1);
    }
}

class DecThread implements Runnable{
    Var v;
    Thread t;

    public DecThread(Var v , String threadName) {
        this.v = v;
        t = new Thread(this , threadName);
    }

    public void run() {
        v.dec(1);
    }
}

public class Prog4_1 {
    public static void main(String[] args) {
        Var v = new Var();
        IncThread i1 = new IncThread(v , "incThread1");
        IncThread i2 = new IncThread(v , "incThread2");
        DecThread d1 = new DecThread(v , "decThread1");
        DecThread d2 = new DecThread(v , "decThread2");

        i1.t.start();
        i2.t.start();
        d1.t.start();
```

```
        d2.t.start();

        try {
            i1.t.join();
            i2.t.join();
            d1.t.join();
            d2.t.join();
        }
        catch(InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("Value after all threads finished : " + v.n);
    }
}
```

Output:



Q2. Repeat Question 1 with shared variables.

Source Code:

```
class Var {
    int n;

    public Var() {
        n = 0;
    }
```

```
    public synchronized void inc(int delta) {
        n += delta;
        System.out.println("Value after increment : " + n);
    }

    public synchronized void dec(int delta) {
        n -= delta;
        System.out.println("Value after decrement : " + n);
    }
}

class IncThread implements Runnable{
    Var v;
    Thread t;

    public IncThread(Var v , String threadName) {
        this.v = v;
        t = new Thread(this , threadName);
    }

    public void run() {
        v.inc(1);
    }
}

class DecThread implements Runnable{
    Var v;
    Thread t;

    public DecThread(Var v , String threadName) {
        this.v = v;
        t = new Thread(this , threadName);
    }
```

```
    public void run() {
        v.dec(1);
    }
}

public class Prog4_2 {
    public static void main(String[] args) {
        Var v = new Var();
        IncThread i1 = new IncThread(v , "incThread1");
        IncThread i2 = new IncThread(v , "incThread2");
        DecThread d1 = new DecThread(v , "decThread1");
        DecThread d2 = new DecThread(v , "decThread2");

        i1.t.start();
        i2.t.start();
        d1.t.start();
        d2.t.start();

        try {
            i1.t.join();
            i2.t.join();
            d1.t.join();
            d2.t.join();
        }
        catch(InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("Value after all threads finished : " + v.n);
    }
}
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 4(git:master))x> javac s4q2.java
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 4(git:master))✓> java s4q2
Value after increment : 1
Value after decrement : 0
Value after decrement : -1
Value after increment : 0
Value after all threads finished : 0
(soumalya|~/Desktop/MotherFolder/Assignments-3rd-sem/Java/Assignment 4(git:master))✓>
```

Q3. Reader writer problem.

Source Code:

```
class Id {
    int id = 0;

    synchronized int assignId() {
        id++;
        return id;
    }
}

class Value {
    int num = 0;
    int activeReaders = 0;
    boolean activeWriter = false;

    synchronized void decRead() {
        activeReaders--;
    }

    synchronized void incRead() {
        activeReaders++;
    }

   void activateWriter() {
```

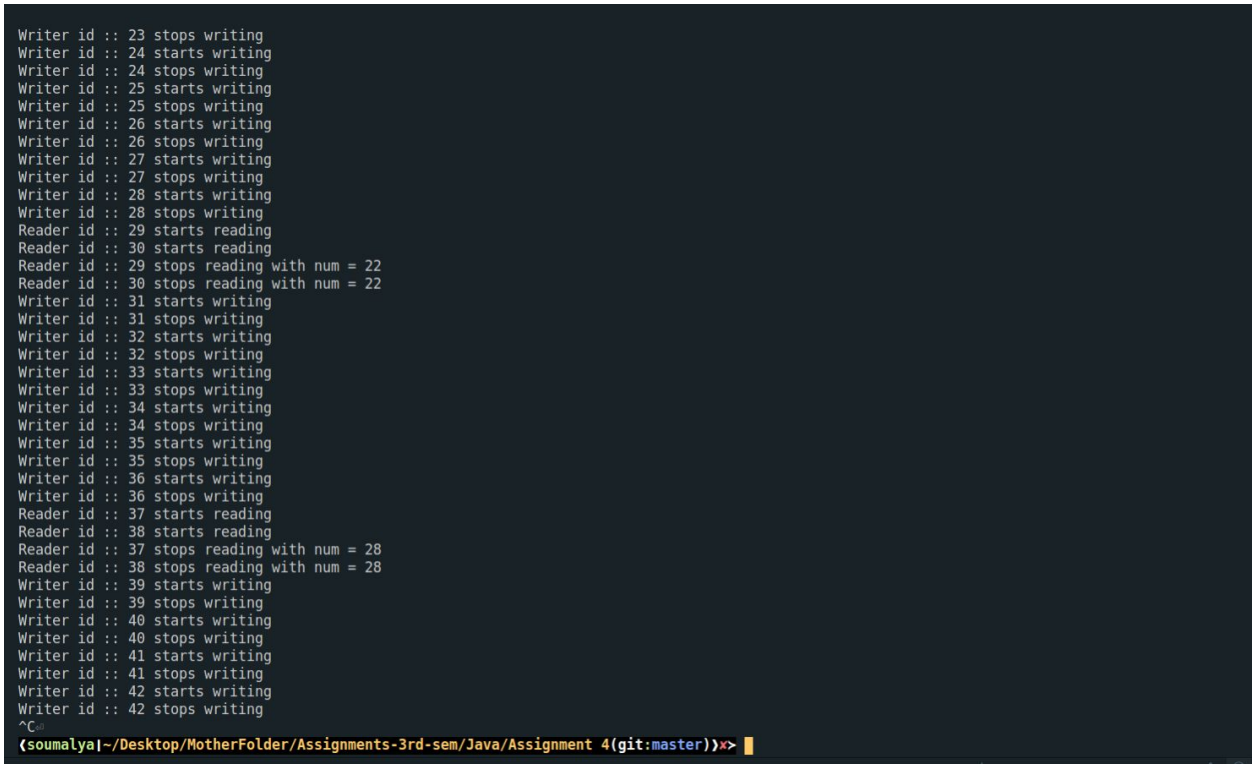
```
        activeWriter = true;
    }

    void deactivateWriter() {
        activeWriter = false;
    }

    public void read(int id) {
        synchronized(this) {
            while(activeWriter == true) {
                try {
                    wait();
                } catch (InterruptedException e) {

                }
            }
        }

        incRead();
        System.out.println("Reader id :: " + id + " starts reading");
        System.out.println("Reader id :: " + id + " stops reading with num = " +
num);
        decRead();
    }

    synchronized void write(int id) {
        while(activeReaders > 0) {
            try {
                wait();
            } catch (InterruptedException e) {

            }
        }

        activateWriter();
```

```
        System.out.println("Writer id :: " + id + " starts writing");
        num++;
        System.out.println("Writer id :: " + id + " stops writing");
        deactivateWriter();
    }
}

class Reader extends Thread {
    Value n;
    Id id;

    public Reader(Value n, Id id) {
        this.n = n;
        this.id = id;
    }

    public void run() {
        while(true) {
            try {
                Thread.sleep((int)(Math.random()%5000 + 3000));
            } catch (InterruptedException e) {

            }

            n.read(id.assignId());
        }
    }
}

class Writer extends Thread {
    Value n;
    Id id;

    public Writer(Value n, Id id) {
        this.n = n;
```

```
        this.id = id;
    }

    public void run() {
        while(true) {
            try {
                Thread.sleep((int)(Math.random())%2000 + 1000);
            } catch (InterruptedException e) {

            }

            n.write(id.assignId());
        }
    }
}

public class Prog4_3 {
    public static void main(String args[]) {
        Value n = new Value();
        Id id = new Id();
        Reader reader1 = new Reader(n, id);
        Reader reader2 = new Reader(n, id);
        Writer writer1 = new Writer(n, id);
        Writer writer2 = new Writer(n, id);

       reader1.start();
       reader2.start();
       writer1.start();
       writer2.start();

       try {
        reader1.join();
        reader2.join();
        writer1.join();
        writer2.join();
```

```
        } catch(InterruptedException e) {

        }
    }
}
```

Output:



Q4. Employee problem.

Source Code:

```
import java.util.*;
import java.awt.event.ActionListener;
import java.io.*;
import javax.swing.*;

class Employee
```

```
{
    String emp_code;
    String name;
    String basic_salary;
    String grade;
    String dept;
    public Employee(String s1,String s2,String s3,String s4,String s5)
    {
        emp_code=s1;
        name=s2;
        basic_salary=s3;
        grade=s4;
        dept=s5;
    }
    public Employee(String s1)
    {
        emp_code=s1;
    }
    public boolean equals(Object t)
    {
        if(t instanceof Employee)
        {
            Employee s1=(Employee)t;
            return emp_code.equals(s1.emp_code);


        }
        else{
            return false;
        }
    }
    public String toString()
    {
```

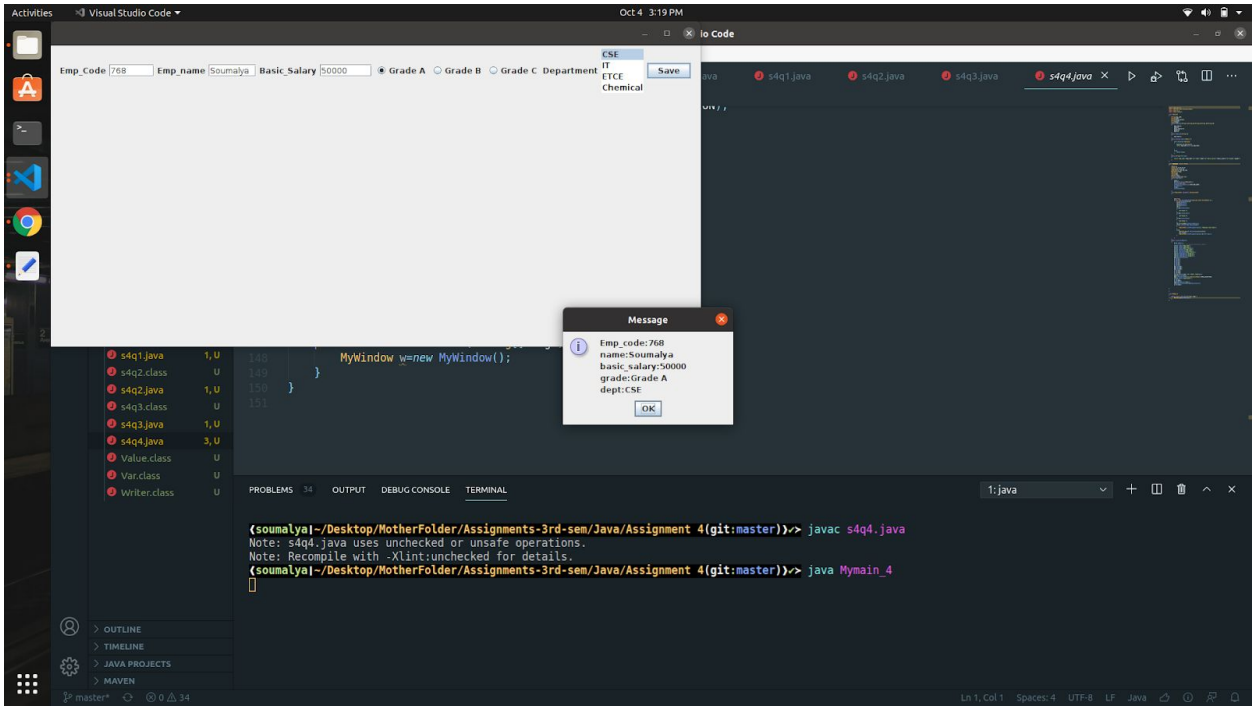
```
        return
"Emp_code:"+emp_code+"\n"+"name:"+name+"\n"+"basic_salary:"+basic_
salary+"\n"+"grade:"+grade+"\n"+"dept:"+dept;
    }
}

class MyWindow extends JFrame
{
    JPanel p;
    JLabel l1,l2,l3,l4,l5;
    JTextField t1,t2,t3;
    JRadioButton rb1,rb2,rb3;
    ButtonGroup bg1;
    JButton b;
    JList dept;
    JButton save;
    ArrayList<Employee> arr;
    public MyWindow()
    {
        super();
        arr=new ArrayList<Employee>();
        setSize(1000,500);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        buildPanel();
        add(p);
        setVisible(true);

    }
    class MyListener implements ActionListener
    {



                @Override
                public void actionPerformed(java.awt.event.ActionEvent e) {
```

```
        String s1,s2,s3,s4,s5;
        s1=t1.getText();
        s2=t2.getText();
        s3=t3.getText();
        s4="";
        if(rb1.isSelected())
        {
           s4="Grade A";
        }
        if(rb2.isSelected())
        {
           s4="Grade B";
        }
        if(rb3.isSelected())
        {
           s4="Grade C";
        }
        s5=(String)dept.getSelectedValue();
        if(arr.contains(new Employee(s1)))
        {
           JOptionPane.showMessageDialog(null,"Employee code taken");
        }
        else{
           Employee e1=new Employee(s1,s2,s3,s4,s5);
           arr.add(e1);
           JOptionPane.showMessageDialog(null,e1.toString());
        }

            }
}
public void buildPanel()
{
   p=new JPanel();
   //p.setLayout(new BoxLayout(p, BoxLayout.Y_AXIS));
   l1=new JLabel("Emp_Code");
```

```
        l2=new JLabel("Emp_name");
        l3=new JLabel("Basic_Salary");
        t1=new JTextField("Emp_code");
        t2=new JTextField("Emp_name");
        t3=new JTextField("Basic Salary");
        rb1=new JRadioButton("Grade A");
        rb2=new JRadioButton("Grade B");
        rb3=new JRadioButton("Grade C");
        bg1=new ButtonGroup();
        p.add(l1);
        p.add(t1);
        p.add(l2);
        p.add(t2);
        p.add(l3);
        p.add(t3);
        bg1.add(rb1);
        bg1.add(rb2);
        bg1.add(rb3);
        p.add(rb1);
        p.add(rb2);
        p.add(rb3);
        String arr[]={"CSE","IT","ETCE","Chemical"};
        dept=new JList(arr);
        dept.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        l4=new JLabel("Department");
        p.add(l4);
        p.add(dept);
        save=new JButton("Save");
        save.addActionListener(new MyListener());
        p.add(save);


}

}
```

```
class Mymain_4
{
    public static void main(String[] args) {
        MyWindow w=new MyWindow();
    }
}
```

Output:



Soumalya Kundu
001810501033

Python Lab Assignment - 1

1. Write a prime generator program using only primes and using python loops.

Source Code:

```
def primeGenerator(n):

    # this function will generate primes upto n
    i = 2
    prime = []
    while i <= n:
        if isPrime(i,prime):
            prime.append(i)
        if(i%2 == 0): i+=1
        else: i += 2

    print(prime)


def isPrime(n, prime):
    if not len(prime):
        #this is only the case when the number is 2
        return True
    else:
        for i in prime:
            # checking if the current number is divisable by already
            # found primes or not
            if n%i == 0:
                return False

    # true returned if the number is not divisable by any of these number
    return True
```

```
if __name__ == "__main__":

    n = int(input("Enter a number: "))
    print("Prime Numbers less than or equal to {}:".format(n))
    primeGenerator(n)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)-> python3 s1q1.py
Enter a number: 50
Prime Numbers less than or equal to 50:
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)->
```

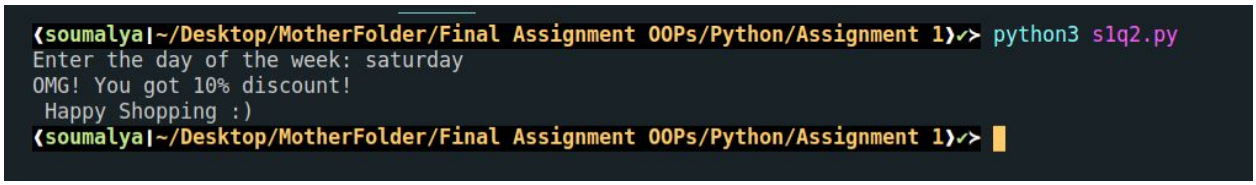
2. Write a discount coupon code using dictionary in Python with different rate coupons for each day of the week.

Source Code:

```
myDict = {
    "sunday" : 10,
    "monday" : 15,
    "tuesday" : 5,
    "wednesday" : 55,
    "thursday" : 34.75,
    "friday" : 21,
    "saturday" : 100
}


if __name__ == "__main__":
```

```
    day = str(input("Enter the day of the week: "))
    day = day.lower()
    if day in myDict.keys():
        print("OMG! You got {}% discount!\n Happy Shopping
:)".format(myDict[day]))
    else:
        print("I am not sure this day exists!:( \nTry again!!!")
```

Output:



3. Print first 10 odd and even numbers using iterators and compress.

Source Code:

```
if __name__ == "__main__":

    odd = []
    even = []

    for num in range(21):
        if num%2 == 0:
            even.append(num)
        else:
            odd.append(num)

    print("first 10 odd numbers: " + str(odd))
    print("first 10 even numbers: " + str(even))
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)-> python3 s1q3.py
first 10 odd numbers: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
first 10 even numbers: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)->
```

4. Print the permutations of ABCDE using iterators.

Source Code:

```
from itertools import permutations

def printPermutation(word):
    p = permutations(word)
    for i in list(p):
        str = ""
        str = str.join(i)
        print(str)

if __name__ == "__main__":
    word = "ABCD"
    newWord = list(word) # just because strings are immutable in python
    n = len(word)
    printPermutation(newWord)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)✓> python3 s1q4.py
ABCD
ABDC
ACBD
ACDB
ADBC
ADCB
BACD
BADC
BCAD
BCDA
BDAC
BDCA
CABD
CADB
CBAD
CBDA
CDAB
CDBA
DABC
DACB
DBAC
DBCA
DCAB
DCBA
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)✓>
```

5. Write a matrix multiplication function to compute.

Source Code:

```
def matrixMultiplication(mat1, mat2):

    row1 = len(mat1)
    col1 = len(mat1[0])

    row2 = len(mat2)
    col2 = len(mat2[0])

    if col1 != row2:
        print("matrix multiplication cannot be performed")
    elif row1 == 0 or col1 == 0 or row2 == 0 or col2 == 0 :
        print("bad input! Multiplication cannot be performed")
    else:

        #initializing the result matrix
```

```
    result = [[0 for i in range(col2)] for j in range(row1)]

    for i in range(row1):
        for j in range(col2):
            for k in range(row2):
                result[i][j] += mat1[i][k] * mat2[k][j]

    for row in result:
        print(row)


if __name__ == "__main__":

    mat1 = [
        [1,2,3],
        [4,5,6],
        [7,8,9]
    ]

    mat2 = [
        [1,2],
        [3,4],
        [5,6]
    ]

    matrixMultiplication(mat1, mat2)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)✓> python3 s1q5.py
[-22, 364]
[-152, 2048]
[-46, 736]
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)✓>
```

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)✓> python3 s1q5.py
matrix multiplication cannot be performed
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)✓>
```

The second result is due to some unauthorised input.

6. Create list of servers, IP addresses and ports using variable positional and keyword arguments.

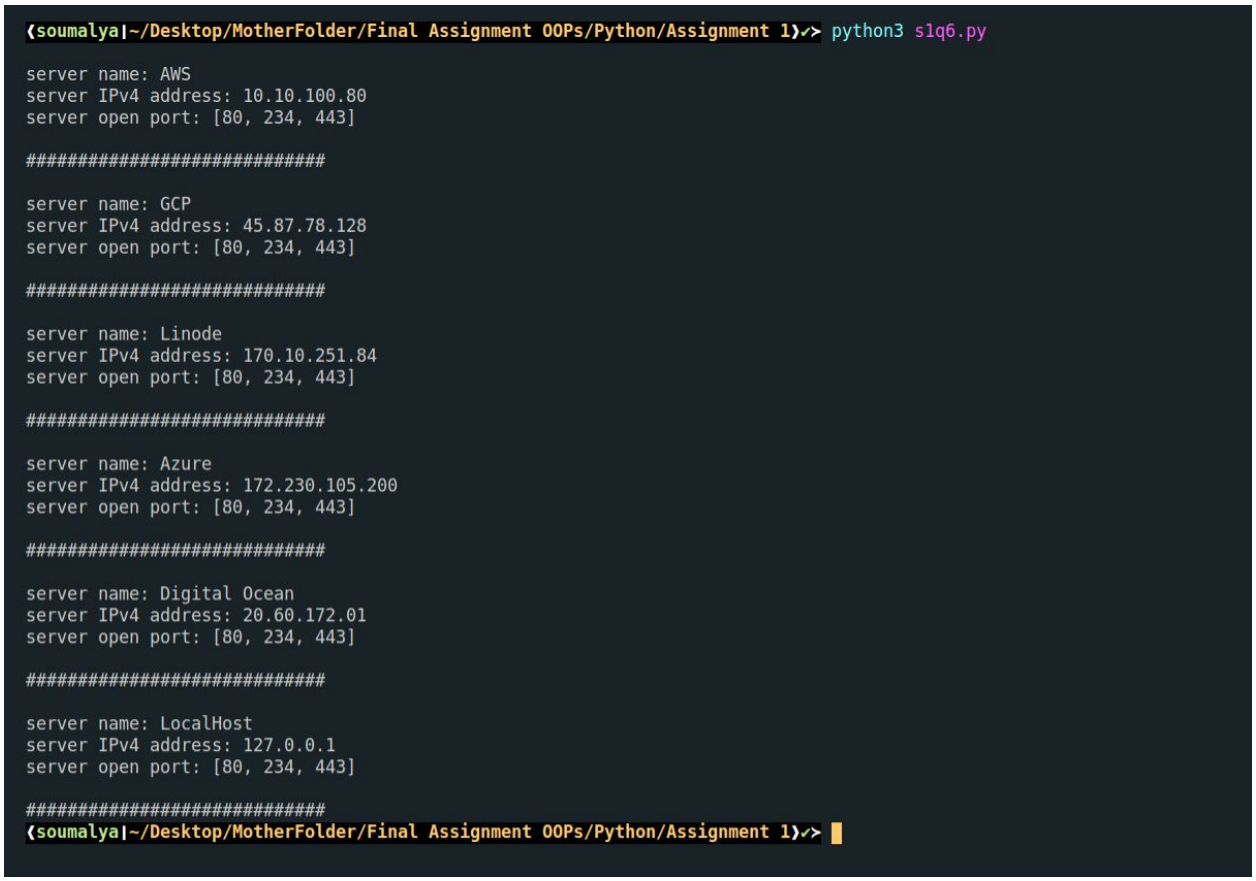
Source Code:

```
myServer = {
   "server1" :
   {
      "name" : "AWS",
      "IPv4" : "10.10.100.80",
      "port" : [80,234,443]
   },

   "server2":
   {
      "name" : "GCP",
      "IPv4" : "45.87.78.128",
      "port" : [80,234,443]
   },
```

```
    "server3" :
    {
        "name" : "Linode",
        "IPv4" : "170.10.251.84",
        "port" : [80,234,443]
    },

    "server4" :
    {
        "name" : "Azure",
        "IPv4" : "172.230.105.200",
        "port" : [80,234,443]
    },

    "server5" :
    {
        "name" : "Digital Ocean",
        "IPv4" : "20.60.172.01",
        "port" : [80,234,443]
    },

    "server6" :
    {
        "name" : "LocalHost",
        "IPv4" : "127.0.0.1",
        "port" : [80,234,443]
    }
}

# this function just shows the all server details
def printServers(**kwargs):

    for server, details in kwargs.items():
        print("\nserver name: {}".format(details["name"]))
```

```
        print("server IPv4 address: {}".format(details["IPv4"]))
        print("server open port: {}".format(details["port"]))
        print("\n##############################")

if __name__ == "__main__":

    printServers(**myServer)
```

Output:



7. Compute sorted five numbers using keyword-only arguments.

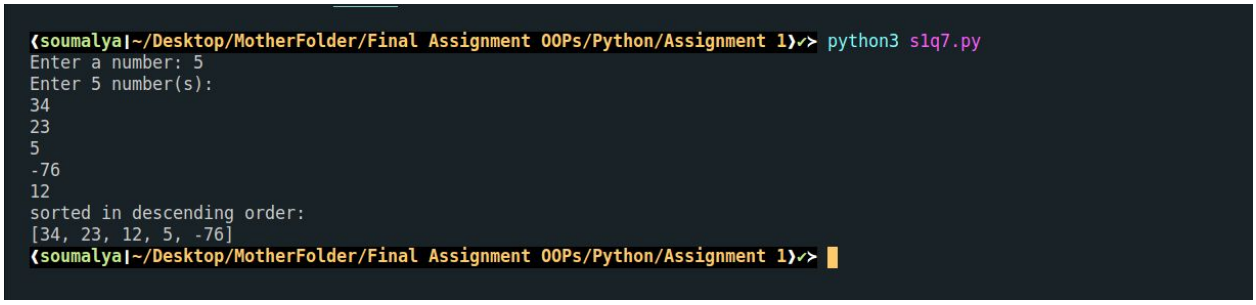
Source Code:

```
def mySort(**kwargs):
```

```
    # this function sorts a list of integers
    myList = []
    sortType = False # False for ascending order sorting
    count = 0
    for key, value in kwargs.items():
        if count == 0:
            myList = value
            count += 1
        else: sortType = value

    n = len(myList)
    #simple bubble sort
    for i in range(n-1):
        for j in range(0, n-i-1):
            if myList[j] > myList[j+1]:
                myList[j], myList[j+1] = myList[j+1],myList[j]

    # reverse the string if the sortType bit is set
    if sortType: myList.reverse()
    if sortType: print("sorted in descending order: ")
    else: print("Sorted in ascending order: ")
    print(myList)


if __name__ == "__main__":

    myList = []
    n = int (input("Enter a number: "))
    print("Enter {} number(s): ".format(n))

    for i in range(0,n):
        x = int(input())
        myList.append(x)

    kwargs = {"list":myList, "type": True}
```

```
mySort(**kwargs)
```

Output:

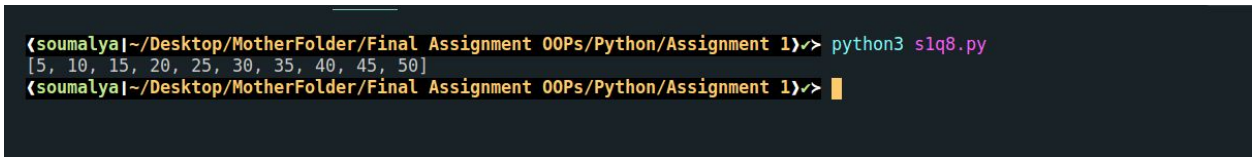


8. Create a list of all the numbers up to N=50 which are multiples of five using anonymous function.

Source Code:

```
if __name__ == "__main__":

    myList = []
    myDivisor =  lambda x,y: (x%y == 0) # generalized lambda function
    for i in range(1,51):
        if myDivisor(i,5):
            myList.append(i)

    print(myList)
```

Output:

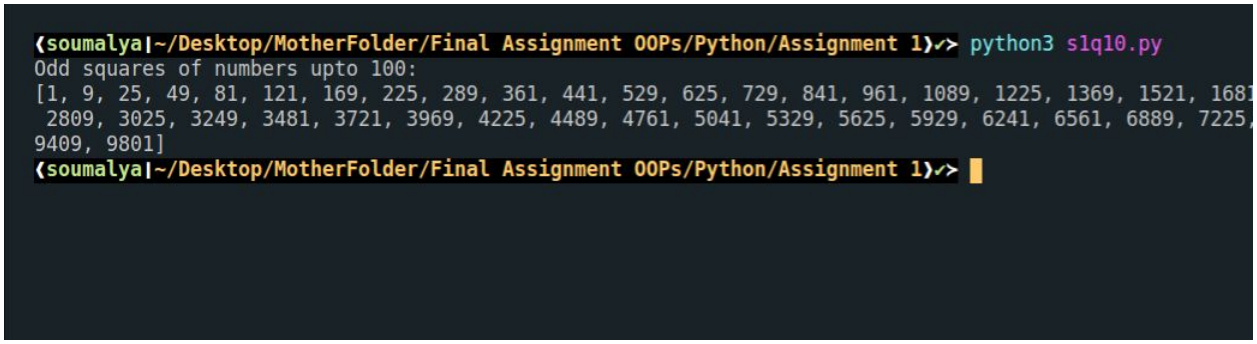


10. Filter out the odd squares using map, filter, list.

Source Code:

```
if __name__ == "__main__":


    #Printing odd squares upto 10000

    ans = list(map( lambda x:x**2 , filter(lambda y:y%2==1,range(100))))
    print(ans)
```

Output:

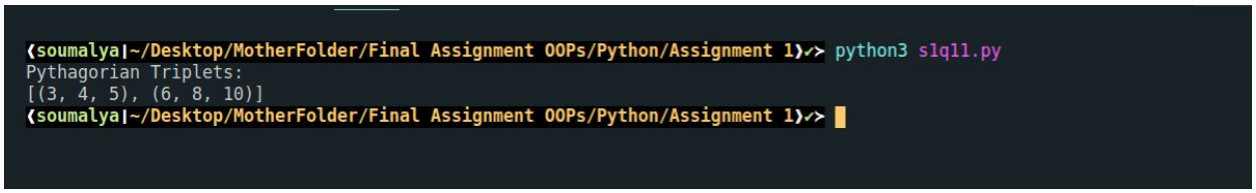


11. Let's find all Pythagorean triples whose short sides are numbers smaller than 10. use filter and comprehension.

Source Code:

```
if __name__ == "__main__":


    res = [(i,j,k) for i in range(1,10) for j in range(i,10) for k in range(j,20)]
#Generating all triplets in the given range
    ans = list(filter(lambda x: x[0]*x[0] +x[1]*x[1]==x[2]*x[2] , res)) #Filtering
the Pythagorean triplets
    print(ans)
```

Output:



12. Enumerate the sequence of all lowercase ASCII letters, starting from 1, using enumerate.

Source Code:

```
if __name__ == "__main__":

    for index,element in enumerate(range(97,123),1):
        print(index,chr(element))
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)-> python3 s1q12.py
1 a
2 b
3 c
4 d
5 e
6 f
7 g
8 h
9 i
10 j
11 k
12 l
13 m
14 n
15 o
16 p
17 q
18 r
19 s
20 t
21 u
22 v
23 w
24 x
25 y
26 z
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)->
```

13. Create a dictionary with comprehension with keys = the letters in the string of your name, and values of the same letters, but with the case swapped.

Source Code:

```
if __name__ == "__main__":

    dic = {x: x.swapcase() for x in "JohnDoe"} #Insert your name here
    print(dic)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)-> python3 s1q13.py
{'S': 's', 'o': 'O', 'u': 'U', 'm': 'M', 'a': 'A', 'l': 'L', 'y': 'Y', 'K': 'k', 'n': 'N', 'd': 'D'}
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 1)->
```

14. Write a python program to

1. read lines from a file, break into tokens and convert the tokens to unique numerical values using python dictionary.

Source Code:

```
if __name__ == "__main__":


    L = ["Dummy lines\n", "oo asas jef lasslakas\n", "paosask jn aslas\n"]

    # writing to file
    file1 = open('myfile.txt', 'w')
    file1.writelines(L)
    file1.close()

    # Using readlines()
    file1 = open('myfile.txt', 'r')
    lines = file1.readlines()


    dic={}
    count=0

    for line in lines:
        words = line.strip().split()
        for word in words:
            if word not in dic:
                dic[word]=count
                count+=1

    print(dic)
```

2. Convert lines of different lengths into lines of same length (maximum length). Use padding if and when required.

Source Code:

```
if __name__ == "__main__":


    L = ["Dummy lines\n", "oo asas jef lasslakas\n", "paosask jn aslas\n"]

    # writing to file
    file1 = open('myfile.txt', 'w')
    file1.writelines(L)
    file1.close()

    # Using readlines()
    file1 = open('myfile.txt', 'r')
    lines = file1.readlines()


    mx=0

    for line in lines:
        if len(line)>mx:
            mx=len(line)
    mx-=1

    ans=[]
    for line in lines:
        rem = mx-len(line)+1
        count = len(line.split())
```

```
    per = rem//(count-1)
    per2 = rem % (count-1)
    st=""
    words = line.split()
    for word in words:
        st+=word
        st+=' '
        for i in range(per):
            st+=' '
        if per2>0:
            st+=' '
            per2-=1
    st.strip()
    st+='\n'
    ans.append(st)

file1.close()

file2 = open('mynewfile.txt', 'w')
file2.writelines(ans)
file2.close()

file2 = open('mynewfile.txt', 'r')
lines = file2.readlines()

for line in lines:
    print(line.strip())
```

15. Write a python program to identify and extract numerical chunks from a text file and convert them into words; e.g.; 1992 “nineteen hundred and ninety two”.

Source Code:

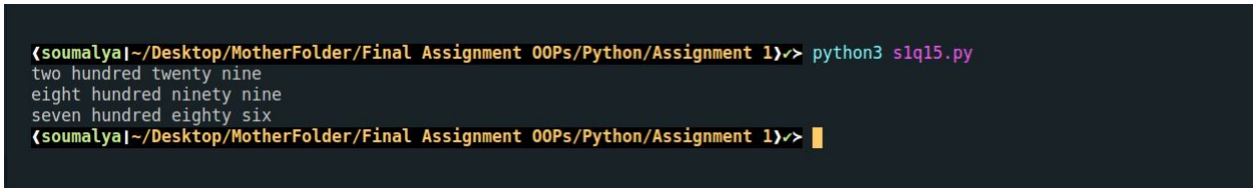
```
def convert_to_words(num):

    # Get number of digits
    # in given number
    l = len(num);

    # Base cases
    if (l == 0):
        print("empty string");
        return;

    if (l > 4):
        print("Length more than 4 is not supported");
        return;

    # The first string is not used,
    # it is to make array indexing simple
    single_digits = ["zero", "one", "two", "three",
                     "four", "five", "six", "seven",
                     "eight", "nine"];

    # The first string is not used,
    # it is to make array indexing simple
    two_digits = ["", "ten", "eleven", "twelve",
                  "thirteen", "fourteen", "fifteen",
                  "sixteen", "seventeen", "eighteen",
                  "nineteen"];

    # The first two string are not used,
    # they are to make array indexing simple
    tens_multiple = ["", "", "twenty", "thirty", "forty",
                     "fifty", "sixty", "seventy", "eighty",
                     "ninety"];
```

```
tens_power = ["hundred", "thousand"];

# Used for debugging purpose only
#print(num, ":", end = " ");

# For single digit number
if (l == 1):
    print(single_digits[ord(num[0]) - '0']);
    return;

# Iterate while num is not '\0'
x = 0;
while (x < len(num)):

    # Code path for first 2 digits
    if (l >= 3):
        if (ord(num[x]) - 48 != 0):
            print(single_digits[ord(num[x]) - 48],
                                  end = " ");
            print(tens_power[l - 3], end = " ");
            # here len can be 3 or 4

        l -= 1;

    # Code path for last 2 digits
    else:

        # Need to explicitly handle
        # 10-19. Sum of the two digits
        # is used as index of "two_digits"
        # array of strings
        if (ord(num[x]) - 48 == 1):
            sum = (ord(num[x]) - 48 +
                   ord(num[x+1]) - 48);
```

```
            print(two_digits[sum]);
            return;

        # Need to explicitely handle 20
        elif (ord(num[x]) - 48 == 2 and
              ord(num[x + 1]) - 48 == 0):
            print("twenty");
            return;

        # Rest of the two digit
        # numbers i.e., 21 to 99
        else:
            i = ord(num[x]) - 48;
            if(i > 0):
                print(tens_multiple[i], end = " ");
            else:
                print("", end = "");
            x += 1;
            if(ord(num[x]) - 48 != 0):
                print(single_digits[ord(num[x]) - 48]);
    x += 1;

if __name__ == "__main__":

    L = ["Dummy lines\n", "19\n", "221 B\n", "paosask jn 47 aslas\n"]

    # writing to file
    file1 = open('myfile.txt', 'w')
    file1.writelines(L)
    file1.close()

    # Using readlines()
    file1 = open('myfile.txt', 'r')
    lines = file1.readlines()
```

```
numbers=[]

for line in lines:
    words = line.strip().split()
    for word in words:
        if word.isdigit():
            numbers.append(word)


for num in numbers:
    convert_to_words(num)
```

Output:



Soumalya Kundu
001810501033

Python Lab Assignment - 2

2. Write first seven Fibinacci numbers using generator next function/ yield in python.

Source Code:

```
# Function to get the next n
# Fibonacci numbers

def next(n):
    fn   = 0
    fn_1 = 1

    while (n > 0):
        yield fn

        tmp = fn_1
        fn_1 = fn + fn_1
        fn = tmp

        n -= 1

# print the next 7
# fibonacci numbers
for i in next(7) :
    print(i)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓> python3 s2q2.py
0
1
1
2
3
5
8
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓>
```

3. Write a code which yields all terms of the geometric progression a, aq, aq2 , aq3 , ... When the progression produces a term that is greater than 100,000, the generator stops (with a return statement). Compute total time and time within the loop.

Source Code:

```
# Generate GP sequence as
# long as a term is < 10^5
def GP(a, q):
    while a < 100000 :
        yield a
        a = a*q

    # Print when the sequence
    # has completed execution
    print("Sequence complete")

# Take values of a & q
# as input
a = int(input("a = "))
q = int(input("q = "))

# Print the sequence
```

```
for i in GP(a, q):
    print(i, end=", ")

print("\b\b\n")
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓> python3 s2q3.py
a = 5
q = 4
5, 20, 80, 320, 1280, 5120, 20480, 81920, Sequence complete
```

4. Create a generator expression for first 10 cubes.

Source Code:

```
# Generate the first n cubes

def getCube(n):
    for i in range(n):
        yield (i+1)*(i+1)*(i+1)

# Print the first 10 cubes
for i in getCube(10):
    print(i, end=", ")

print("\b\b")
```

Output:

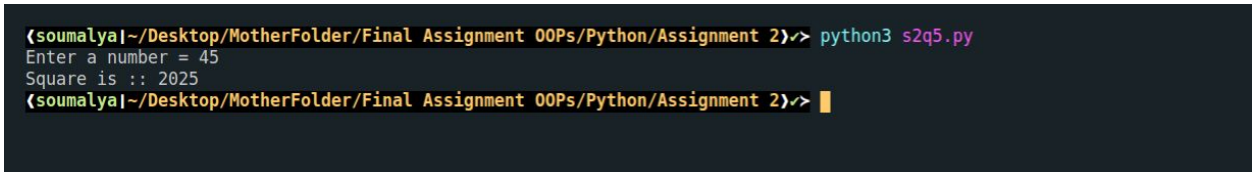
```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓> python3 s2q4.py
First 10 cube numbers are:
1, 8, 27, 64, 125, 216, 343, 512, 729, 1000,
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓>
```

5. Write a program to compute square area of square class with self() to get square value in python.

Source Code:

```
# class square to take a number
# in constructor and find it's
# square.
class square:
    def __init__(self, n):
        self.n = n
    def getSquare(self):
        return self.n*self.n

# Print the square
a = int(input("a = "))
sq = square(a)

print("Square is ::", sq.getSquare())
```

Output:



6. Create book, ebook, journal classes to use inheritance with title, publisher, page, year of publishing details.

Source Code:

```
# parent class :: book
class book:
```

```
    def __init__(self, title, pub):
        self.title = title
        self.pub = pub

    def show_det(self) :
        print("Is Book")
        print(self.title, " ", self.pub)

# child class of 'book' :: ebook
# ebook has an additional attr. :: page
class ebook(book):
    def __init__(self, title, pub, page):
        super().__init__(title, pub)
        self.page = page

    def show_det(self):
        print("Is EBook")
        print(self.title, " ",self.pub, " ",self.page)

# child class of 'book' :: journal
# journal has an additional attr. :: page
class journal(book):
    def __init__(self, title, pub, page):
        super().__init__(title, pub)
        self.page = page

    def show_det(self):
        print("Is Journal")
        print(self.title, " ",self.pub, " ",self.page)


# Creating the classes and using them
new_book = book("Fault in our Stars", "John Green")
new_book.show_det()
```

```
print("-----------")

new_ebook = ebook("Stormlight Archive", "Sanderson", 100)
new_ebook.show_det()

print("-----------")

new_journal = journal("ACM-Programming Languages", "Springer", 3)
new_journal.show_det()
```

Output:

```
(soumalya)~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓> python3 s2q6.py
Is Book
Fault in our Stars   John Green
-----------
Is EBook
Stormlight Archive   Sanderson   100
-----------
Is Journal
ACM-Programming Languages   Springer   3
(soumalya)~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓>
```

7. Show multiple inheritance in shape, 2-D shapes, 3-D shapes, square, rectangle, polygon, hexagon, cube, cone, cylinder etc. classes with their areas.

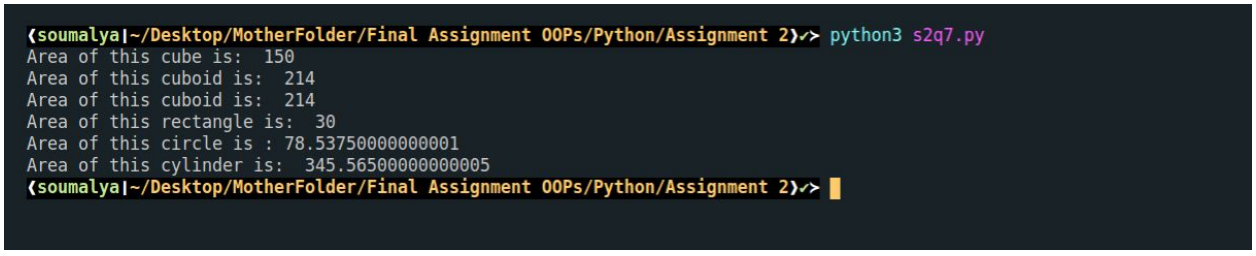
Source Code:

```
class Smooth:
    def getArea(self):
        print("Not Available at this level")
class NonSmooth:
    def getArea(self):
        print("Not Available at this level")
class TwoShape:
    def getArea(self):
        print("Not Available at this level")
class ThreeShape:
    def getArea(self):
```

```
        print("Not Available at this level")
class cuboid(ThreeShape,NonSmooth):
    def __init__(self,l,b,h):
        self.l=l
        self.b=b
        self.h=h
    def getArea(self):
        print("Area of this cuboid is
{}",2*(self.h*self.b+self.l*self.b+self.h*self.l))
class cube(ThreeShape,NonSmooth):
    def __init__(self,a):
        self.a=a
    def getArea(self):
        print("Area of this cube is {}",6*self.a*self.a)
class square(TwoShape,NonSmooth):
    def __init__(self,a):
        self.a=a
    def getArea(self):
        print("Area of this square is {}",self.a*self.a)
class rectangle(TwoShape,NonSmooth):
    def __init__(self,l,b):
        self.l=l
        self.b=b
    def getArea(self):
        print("Area of this rectangle is {}",self.b*self.l)
class circle(TwoShape,Smooth):
    def __init__(self,r):
        self.r=r
    def getArea(self):
        print("Area of this circle is {}",self.r*self.r*3.1415)
class cylinder(ThreeShape,Smooth):
    def __init__(self,r,h):
        self.r,self.h=r,h
    def getArea(self):
```

```
        print("Area of this cylinder is
{}",2*self.r*self.h*3.1415+2*3.1415*self.r*self.r)


cube1 = cube(5)
cube1.getArea()

cuboid1 = cuboid(5,6,7)
cuboid1.getArea()

square1 = square(5)
cuboid1.getArea()

rectangle1 = rectangle(5,6)
rectangle1.getArea()

circle1 = circle(5)
circle1.getArea()

cylinder1 = cylinder(5,6)
cylinder1.getArea()
```

Output:

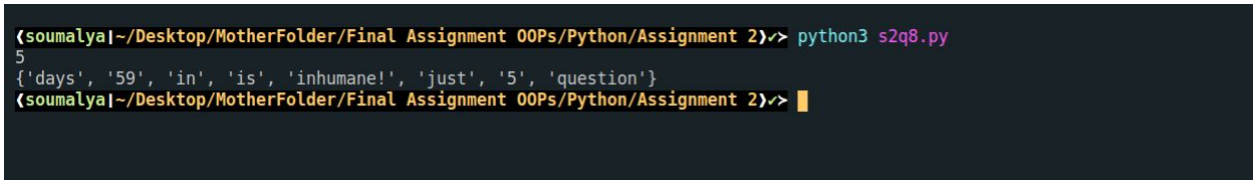


8. Search for palindrome and unique words in a text using class method and string method.

Source Code:

```
class Storage:
    def __init__(self,text):
        self.text=text
    def isPal(self,word):
        for idx in range(0,len(word)-1):
            if word[idx]!=word[len(word)-1-idx]:
                return False
        return True
    def getPalindromes(self):
        words=list(self.text.split())
        for word in words:
            if self.isPal(word):
                print(word,end=' ')
        print('')
    def getUniqueWords(self):
        words=set(self.text.split())
        print(words)

storage=Storage('hello i am a bad boy geek for geek')
storage.getPalindromes()
storage.getUniqueWords()
```

Output:



9. Check and set a person's age in person class using property decorator.

Source Code:

```
class Person:
    def __init__(self):
```

```
        self.__name=''
        self.__age=-1
    @property
    def name(self):
        return self.__name
    @property
    def age(self):
        return self.__age
    @name.setter
    def name(self, value):
        self.__name=value
    @age.setter
    def age(self,value):
        self.__age=value
    @name.deleter
    def name(self):
        print('Deleting name property')
        del self.__name
    @age.deleter
    def age(self):
        print('Deleting Age property')
        del self.__age

p=Person()
p.name='Geek God'
print("Name: ",p.name)
p.age=18
print("Age: ",p.age)
del p.name
del p.age
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓> python3 s2q9.py
Name:  Soumalya Kundu
Age:  4546
Name property deleted
Age property deleted
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)✓>
```

10. Write a operator overloading for “len” which shows string length for any given string and return only length of last three words if the string is in "Hello! I am 42 years old!" format.

Source Code:

```
class MyString:
    def __init__(self,s):
        self.str=s
    def __len__(self):
        words=list(self.str.split())
        if len(words)==1:
            length=0
            for ch in words[0]:
                length+=1
            return length
        else:
            length=0
            count=len(words)
            for ch in words[count-1]:
                length+=1
            if count>=2:
                for ch in words[count-2]:
                    length+=1
            if count>=3:
                for ch in words[count-3]:
                    length+=1
            length+=count-1
            return length
```

```
myString = MyString('Hello i am 42 years old')
print("Length of '{0}' : {1}".format(myString.str,len(myString)))

myString = MyString('Helloiam42yearsold')
print("Length of '{0}' : {1}".format(myString.str,len(myString)))
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)↵> python3 s2q10.py
Length of 'India is now on 210th day of quarantine' : 22
Length of 'But also we have to submit inhuman task' : 24
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)↵>
```

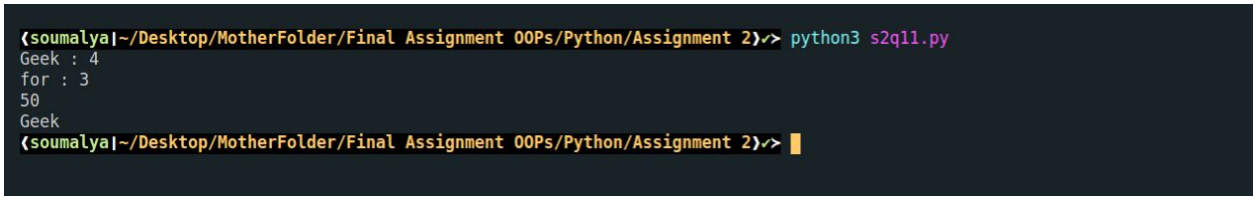
11. Write a operator overloading for “len” which shows string length for any given string and return only length of repetitive words with the text if the text has some repetitive parts. Determine the most frequently occurring words using most_common.

Source Code:

```
class MyString:
    def __init__(self,s):
        self.str=s
    def __len__(self):
        length=len(self.str)
        words=list(self.str.split())
        freq = {}
        for item in words:
            if item in freq:
                freq[item] += 1
            else:
                freq[item] = 1
        for key,value in freq.items():
            if value>1:
```

```
            print("{0} : {1}".format(key,len(key)))
        return length
    def most_common(self):
        words=list(self.str.split())
        freq = {}
        for item in words:
            if item in freq:
                freq[item] += 1
            else:
                freq[item] = 1
        ans,f="",-1
        for key,value in freq.items():
            if value>f:
                f,ans=value,key
        return ans

myString = MyString('Hello i am a geeks for geeks follower')
print(len(myString))
print(myString.most_common())
```

Output:



12. Write a function that flattens a nested dictionary structure like one obtained from Twitter and Facebook APIs or from some JSON file.

nested = {

'fullname': 'Alessandra',

'age': 41,

'phone-numbers': ['+447421234567', '+447423456789'],

'residence': {

'address': {

'first-line': 'Alexandra Rd',

'second-line': '',

Testing, Profiling, and Dealing with Exceptions

[230]

},

'zip': 'N8 0PP',

'city': 'London',

'country': 'UK',

},

}

Source Code:

```
# To flatten the dict, we've to
# recursively traverse and check
# if there's a key : dict mapping
# if so, fix that
# if not return a prefix : d

def flatten(d, prefix=""):
    if isinstance(d, dict):
        newDict = {}
```

```
        for k, v in d.items():
            getDict = flatten(v, prefix+"_"+k if prefix != "" else k)
            for kk, vv in getDict.items():
                newDict[kk] = vv
        return newDict
    else :
        return {prefix: d}

# example dict
mydict = {
    "fullname" : "Alessandra",
    "age" : 41,
    "phone_numbers" : ["+447421234", "+9910938383"],
    "residence" : {
        "address" : "Alexindra Road",
        "second_line" : "Nothing Special",
        "zip" : "YOHERE",
        "city" : "London",
        "country" : "UK"
    }
}

# Flatten the above dict
newdict = flatten(mydict)

# Print the flattened dict
print(newdict)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)➢> python3 s2q12.py
{'fullname': 'Alessandra', 'age': 41, 'phone_numbers': ['+447421234', '+9910938383'], 'residence_address': 'Alexindra Road', 'residence_second_line': '
Nothing Special', 'residence_zip': 'YOHERE', 'residence_city': 'London', 'residence_country': 'UK'}
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)➢>
```

13. Use parameterized or nose_parameterized to compute power of following values:

(2, 2, 4),

(2, 3, 8),

(1, 9, 1),

(0, 9, 0). Use pytest to check errors.

Source Code:

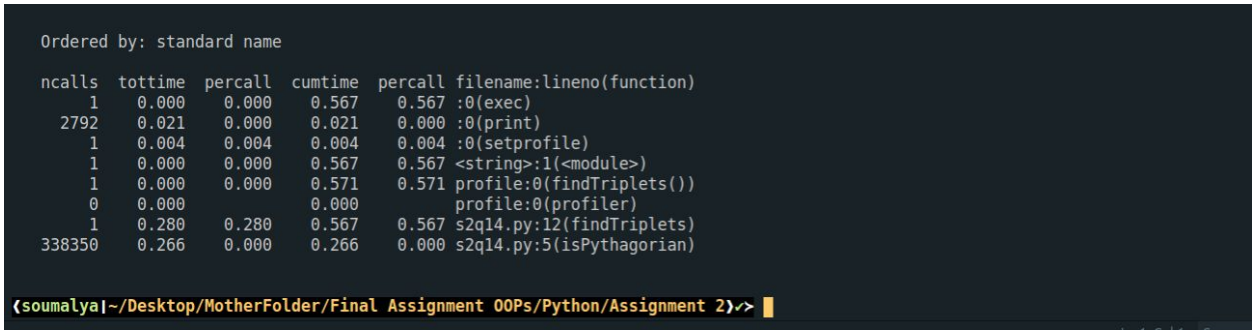
```
from nose.tools import assert_equal

from parameterized import parameterized

import math

# Values in the form
# (x, y, expected after finding x^y)
@parameterized ([
    (2, 2, 4),
    (2, 3, 8),
    (1, 9, 1),
    (0, 9, 0),
    (2, 4, 11)
])

# function checking whether the test
# outputs are as expected
def test_pow(base, exponent, expected):
    assert_equal(math.pow(base, exponent), expected)

# to run this program use
# nosetests -v PLA2_13.py
```

14. Use profile/cprofile to check pythogorian triples code in python. Think about time complexity of the code.

Source Code:

```
import profile

# Function to check is a
# triplet is Pythagorian Triplet
def isPythagorian(a, b, c):
    if a^2 + b^2 == c^2 :
        return True
    return False

# Function to iterate over
# values of a, b & c
def findTriplets():
    for i in range(100):
        for j in range(i+1):
            for k in range(i+1):
                if isPythagorian(i, j, k):
                    print(i, j, k)

# Python profiler
profile.run('findTriplets()')
```

Output:



15. Write a program to sort in descending order by the sum of credits accumulated by students, so to have the best student at position 0. Write a function using map, to produce a decorated object, to sort, and then to undecorate. Each student has credits in three (possibly different) subjects. To decorate an object means to transform it, either adding extra data to it, or putting it into another object, in a way that allows to sort the original objects the way you want. After the sorting, one reverts the decorated objects to get the original ones from them. This is called to undecorate.

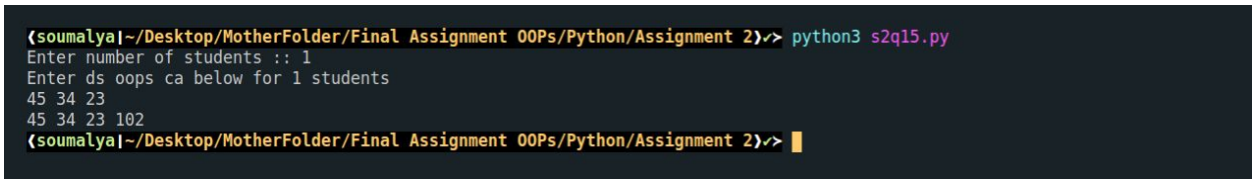
Source Code:

```
class student:
    # ds for Data Structure marks
    # oops for Object Oriented Programming S
    # ca for Computer Architecture
```

```
    def __init__(self, ds, oops, ca) :
        self.ds = ds
        self.oops = oops
        self.ca = ca

    def sum(self):
        return self.ds + self.ca + self.oops

n = int(input("Enter number of students :: "))
print("Enter ds oops ca below for", n, "students")

studlist = []

for i in range(n):
    stud = student(0, 0, 0)
    # Get the marks for a student
    stud.ds, stud.oops, stud.ca = [ int(x) for x in input().split(" ")]
    studlist.append(stud)

# Decorating the object student with
# the sum of the marks
decStudList = []
for stud in studlist :
    decStudList.append((stud, stud.sum()))

# Sorting on the basis of sum of the marks
decStudList.sort(key=lambda tupstud: tupstud[1])

# UnDecorating the list to get the original list
studlist = []

for tupstud in decStudList:
    studlist.append(tupstud[0])

# Printing the sorted list
```

```
for stud in studlist:
    print(stud.ds, stud.oops, stud.ca, stud.sum())
```

Output:



16. Write a python program to calculate the number of editing operations (substitution, deletion and insertion) in the output sequence with respect to a given reference input. Prepare the Minimum Edit Distance (MED) Table and print the backtrace to MED (Consider the root form of words while calculating the number of editing operations)

Source Code:

```
# Calculating Minimum Edit Distance
def calcMED(str1, str2):
    m = len(str1)
    n = len(str2)

    # Creating a 2D array
    dp = [[0 for x in range(n+1)] for x in range(m+1)]


    for i in range(m+1):
        for j in range(n+1):
```

```
        # If no character from string 1 is taken
        # after j operations on string 2
        if i == 0 :
            dp[i][j] = j
        # if no character from string 2 is taken
        # after i operations on string 1
        elif j == 0:
            dp[i][j] = i
        # if the character are same
        elif str1[i-1] == str2[j-1]:
            dp[i][j] = dp[i-1][j-1]
        else :
            # If they are different
            dp[i][j] = 1 + min(dp[i][j-1] # deletion in str1
                       , dp[i-1][j-1],   # substitution
                        dp[i-1][j])      # insertion in str1

cnt = dp[m][n]

backtrace = ""

while m > 0 and n > 0 :
    # if there's no operation
    if dp[m][n] == 0 :
        backtrace += "s"
        m -= 1
        n -= 1
    # if the characters are same
    elif dp[m-1][n-1] == dp[m][n] and str1[m-1] == str2[n-1]:
        backtrace += "s"
        m -= 1
        n -= 1
    # if there's deletion of a character from str1
    elif dp[m-1][n] + 1 == dp[m][n]:
        backtrace += "D"
```

```
            m -= 1
        # if there's insertion of a character in str1
        elif dp[m][n-1] + 1 == dp[m][n]:
            backtrace += "I"
            n -= 1
        # if there's substitution
        else :
            backtrace += "S"
            m -= 1
            n -= 1

    # reverse the backtrace
    backtrace = backtrace[::-1]
    return cnt, backtrace

str1 = input("1st string = ")
str2 = input("2nd string = ")
cnt, backtrace = calcMED(str1, str2)
print(cnt, backtrace)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)-> python3 s2q16.py
1st string = you shall overcome
2nd string = you willl overcome
3 ssssSSSssssssssss
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)->
```

17. Write a single python program to do the following operations on a text file by writing different user defined functions.

a. Remove all the special characters.

b. Remove all single characters.

c. Substitute multiple spaces with single space.

d. Convert all the words into Lowercase.

e. Convert the words into literal form from their contracted form (e.g., Couldn't Could not).

Source Code:

```
## Code is self explainatory

class WordProcessorEngine:
    contractions = {
"ain't": "am not / are not",
"aren't": "are not / am not",
"can't": "cannot",
"can't've": "cannot have",
"'cause": "because",
"could've": "could have",
"couldn't": "could not",
"couldn't've": "could not have",
"didn't": "did not",
"doesn't": "does not",
"don't": "do not",
"hadn't": "had not",
"hadn't've": "had not have",
"hasn't": "has not",
"haven't": "have not",
"he'd": "he had / he would",
"he'd've": "he would have",
"he'll": "he shall / he will",
"he'll've": "he shall have / he will have",
"he's": "he has / he is",
"how'd": "how did",
```

```
"how'd'y": "how do you",
"how'll": "how will",
"how's": "how has / how is",
"i'd": "I had / I would",
"i'd've": "I would have",
"i'll": "I shall / I will",
"i'll've": "I shall have / I will have",
"i'm": "I am",
"i've": "I have",
"isn't": "is not",
"it'd": "it had / it would",
"it'd've": "it would have",
"it'll": "it shall / it will",
"it'll've": "it shall have / it will have",
"it's": "it has / it is",
"let's": "let us",
"ma'am": "madam",
"mayn't": "may not",
"might've": "might have",
"mightn't": "might not",
"mightn't've": "might not have",
"must've": "must have",
"mustn't": "must not",
"mustn't've": "must not have",
"needn't": "need not",
"needn't've": "need not have",
"o'clock": "of the clock",
"oughtn't": "ought not",
"oughtn't've": "ought not have",
"shan't": "shall not",
"sha'n't": "shall not",
"shan't've": "shall not have",
"she'd": "she had / she would",
"she'd've": "she would have",
"she'll": "she shall / she will",
```

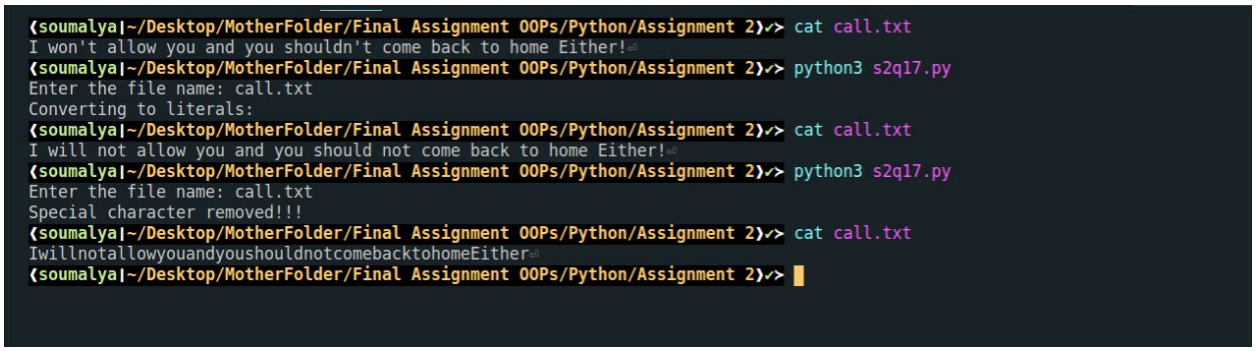
"she'll've": "she shall have / she will have",
"she's": "she has / she is",
"should've": "should have",
"shouldn't": "should not",
"shouldn't've": "should not have",
"so've": "so have",
"so's": "so as / so is",
"that'd": "that would / that had",
"that'd've": "that would have",
"that's": "that has / that is",
"there'd": "there had / there would",
"there'd've": "there would have",
"there's": "there has / there is",
"they'd": "they had / they would",
"they'd've": "they would have",
"they'll": "they shall / they will",
"they'll've": "they shall have / they will have",
"they're": "they are",
"they've": "they have",
"to've": "to have",
"wasn't": "was not",
"we'd": "we had / we would",
"we'd've": "we would have",
"we'll": "we will",
"we'll've": "we will have",
"we're": "we are",
"we've": "we have",
"weren't": "were not",
"what'll": "what shall / what will",
"what'll've": "what shall have / what will have",
"what're": "what are",
"what's": "what has / what is",
"what've": "what have",
"when's": "when has / when is",
"when've": "when have",

```
"where'd": "where did",
"where's": "where has / where is",
"where've": "where have",
"who'll": "who shall / who will",
"who'll've": "who shall have / who will have",
"who's": "who has / who is",
"who've": "who have",
"why's": "why has / why is",
"why've": "why have",
"will've": "will have",
"won't": "will not",
"won't've": "will not have",
"would've": "would have",
"wouldn't": "would not",
"wouldn't've": "would not have",
"y'all": "you all",
"y'all'd": "you all would",
"y'all'd've": "you all would have",
"y'all're": "you all are",
"y'all've": "you all have",
"you'd": "you had / you would",
"you'd've": "you would have",
"you'll": "you shall / you will",
"you'll've": "you shall have / you will have",
"you're": "you are",
"you've": "you have"
}

    @staticmethod
    def convert(word):
        ending_puntucation = ""
        if not word[-1].isalnum():
            ending_puntucation = word[-1]
            word = word[:-1]
```

```
        result = word
        if word.lower() in WordProcessorEngine.contractions:
            result = WordProcessorEngine.contractions[word.lower()]
            if word.lower() != word:
                result = result.capitalize()

        result += ending_puntucation
        return result

class FileWrapper:
    def __init__(self, filename):
        self.filename = filename

    def applyFilter(self, filter):
        f = open(self.filename, "r")
        lines = f.readlines()
        lines = list(map(filter, lines))
        f.close()
        f = open(self.filename, "w")
        f.write('\n'.join(lines))
        f.close()

    def removeSpecialCharacters(self):
        self.applyFilter(lambda x : ''.join(char for char in x if char.isalnum()))

    def removeSingleCharacters(self):
        self.applyFilter(lambda x : ' '.join(word for word in x.split() if len(word)
!= 1))

    def removeMultipleSpaces(self):
        self.applyFilter(lambda x : ' '.join(word for word in x.split()))

    def toLowerCase(self):
        self.applyFilter(lambda x : x.lower().rstrip())
```

```
    def toLiterals(self):
        self.applyFilter(lambda x : ' '.join(WordProcessorEngine.convert(word)
for word in x.split()))



## Tests
filename = input()
f = FileWrapper(filename)
f.removeSpecialCharacters()
f.removeSingleCharacters()
f.removeMultipleSpaces()
f.toLowerCase()
f.toLiterals()
```

Output:



18. Using Numpy create random vector of size 15 having only Integers in the range 0 -20. Write a program to find the most frequent item/value in the vector list.

Source Code:

```
import numpy as np

# Creating a list with random integers
l = [np.random.randint(-20, 11) for x in range(15)]

# Displaying a list with random integers
print(l)

# Storing frequency of integers
freq = {}

for x in l:
    if x in freq:
        freq[x] += 1
    else :
        freq[x] = 1

# Finding the element with max frequency
kf, vf = l[0], freq[l[0]]

for k, v in freq.items():
    if v > vf:
        kf  = k

# Printing the element with max freq.
print(kf)
```

Output:

```
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)-> python3 s2q18.py
[-17, 0, -7, -12, 3, -10, 10, 10, 7, -13, -9, -11, 9, -11, -19]
-11
(soumalya|~/Desktop/MotherFolder/Final Assignment OOPs/Python/Assignment 2)->
```

19.Check http://yann.lecun.com/exdb/mnist/ web page. Execute the training-testing model of

classifications and compare accuracy and other ROC measures for the classification solutons for any

two algorithms using Python among - K-NN with non-linear deformation (IDM), K-NN, shape context

matching, Y. LeCun, L. Bottou and Y. Bengio: Reading Checks with graph transformer networks, 3-layer NN, 500+300 HU, softmax, cross entropy, weight decay, 6-layer NN 784-2500-

2000-1500-1000-500-10 (on GPU) [elastic distortions], Convolutional net Boosted LeNet-4,

[distortions], committee of 35 conv. net, 1-20-P-40-P-150-10 [elastic distortions].

Source Code:

```
import numpy as np
import gzip
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, roc_auc_score
from sklearn.neural_network import MLPClassifier

testLoc={
   'image':'MNIST_DATA/t10k-images-idx3-ubyte.gz',
   'label': 'MNIST_DATA/t10k-labels-idx1-ubyte.gz'
}
trainLoc={
```

```
    'image':'MNIST_DATA/train-images-idx3-ubyte.gz',
    'label': 'MNIST_DATA/train-labels-idx1-ubyte.gz'
}

def extract_Images(filename):
    f=gzip.open(filename,'r')
    f.read(4) #magic number
    nImgs= int.from_bytes(f.read(4),'big')
    row= int.from_bytes(f.read(4),'big')
    col= int.from_bytes(f.read(4),'big')
    data=f.read()
    images=
np.frombuffer(data,dtype=np.uint8).astype(np.float32).reshape((nImgs,row,
col))
    return images

def extract_Labels(filename):
    file = gzip.open(filename,'rb')
    file.read(8)
    data= file.read()
    labs = np.frombuffer(data,dtype=np.uint8).astype(np.int32)
    return labs

trainImg = extract_Images(trainLoc['image'])
trainLab = extract_Labels(trainLoc['label'])

testImg = extract_Images(testLoc['image'])
testLab = extract_Labels(testLoc['label'])

print(trainImg.shape)
print(testImg.shape)
print(trainLab.shape)
print(testLab.shape)

#TRAIN function
```

```
def train(images, labels, classifier):
    images = images.reshape(images.shape[0],-1)
    classifier.fit(images,labels)

    print("Training Done..")

    ''' Training set Performance (but costly ops) '''
    # output = classifier.predict(images)
    # train_acc = (output==labels).sum()/labels.shape[0]
    # print("Training Accuracy: ", train_acc)

#TEST function
def test(images,labels,classifier):
    images = images.reshape(images.shape[0],-1)
    op = classifier.predict(images)
    # train(trainImg,trainLab,NN)

    test_acc = (op==labels).sum()/labels.shape[0]
    print("Test set Accuracy: {:.3f}%".format(test_acc*100))
    clReport= classification_report(labels,op, digits=3)
    print(clReport)
    auc = roc_auc_score(labels, classifier.predict_proba(images),
multi_class="ovr")
    print("ROC AUC Score: ",auc)



KNN = KNeighborsClassifier(n_neighbors=10)

NN = MLPClassifier(random_state=1, solver= 'adam', max_iter=300,
hidden_layer_sizes=(256,128,64))

#NN
print("Three layer NN")
train(trainImg,trainLab,NN)
```

```
test(testImg,testLab,NN)

#Training Testing in KNN classifier
print("KNN Classifier")
train(trainImg,trainLab,KNN)
test(testImg[:500],testLab[:500],KNN)
```

Output:

20. Create a quiz GUI using inheratnce and polymorphism in Python to ask a sequence of

questions of the user. You may follow controller.py, quiz.py or shortAnswer.py files or may write your

own codes. The interface is shown here:

https://cs.calvin.edu/courses/cs/108/vnorman/13oop/lab.html

The application displays problems in a text area (on the top) and messages (on the bottom right); the

user types answers in a text box (on the bottom left) and presses enter.

Three classes work together to create the application:

Controller - implements a GUI driver for a quiz.

Quiz - implements a simple quiz with short answer problems.

o Keeps track of current problem.

o Randomizes the order of the problems.

o Keeps track of whether or not all problems have been used.

ShortAnswer - implements a simple problem class with a string question and string answer.

o Provides interface for asking the question.

o Provides interface for checking if a provided answer is correct.

Exercise 20.1

Create a new package for this lab called Exercise20 and copy the starting code files into this

package: controller.py, quiz.py, shortAnswer.py (files attached).

Familiarize yourself with the quiz mechanism by doing the following:

Run the controller a couple times;

Run the unit tests for the ShortAnswer problem class;

Add a new short-answer problem (of your choice) to the quiz.

Inheritance

Right now this quiz mechanism can only ask short answer problems, but this is a bit too limited for

the purposes. You will add fill-in-the-blank problems, true-false problems, and maybe even multiple

choice problems. As we start planning, we realize that we will be duplicating code if we write each

kind of problem from scratch. Instead, let's start with a Problem class that will be the parent class of

all the different kinds of problems. This class will collect all of the attributes and methods that are
shared between all problems.
So what is shared between all problems? Each problem has some text, but asks the question in a
different way (e.g. short-answer just added a question mark, but a fill-in-the-blank problem should
add both the question mark and an indication to "Fill in the blank."). Further, all problems have
answers, but a true-false problem has a boolean answer instead of a string. Considering these
properties, we proceed as follows:

Exercise 20.2

Refactor your code to include a Problem class and a ShortAnswer class that inherits from Problem.

The new Problem class:

Includes a constructor that receives a string and stores it in an instance variable called self.text

Includes an accessor for the question called get_text()

Put this in a separate file.

The updated ShortAnswer class:

Has an updated class declaration that indicates it is inheriting from Problem

Has an updated constructor:

o Calls the Problem constructor to initialize the text instance variable (instead of doing

the assignment statement itself): Problem.__init__(self, q)

o Does not remove or change the initialization of the answer instance variable

Has an updated ask_question method that replaces the access of self.text with a call to the

appropriate accessor in the Problem class and then appends the question mark: self.get_text() + '?'

Removes the get_text method (since the Problem class is taking care of this for us).

If all has gone well, your controller should run just as it did before, and all of the ShortAnswer

tests should still pass.

Python does not require that the question class definitions be placed in separate files, but it is

common practice to separate more complex classes into separate files. These question classes are

probably simple enough that they can be kept together, but if you choose this option, you should

rename the file to “problem” to more accurately indicate what it contains.

Polymorphism

We are now ready to add more types of problems, such as true-false problems and fill-in-the-blank
problems. Examples of how your application should present these questions are shown below:

The new problem classes are very similar to a ShortAnswer problem, but they each have
differences:
The FillInTheBlank class is very similar to the ShortAnswer class (so similar we should maybe
not have a whole separate class, but this is for learning purposes, right?) with the following
modification:
o The ask_question method must append the string '\nFill in the blank.' to the end of the
problem text instead of just a question mark.

The TrueFalse class is also similar, but with a few more details to be worked out:
o The answer is expected to be a boolean value, so the constructor should raise an exception if the calling program provides an answer that is not an instance of bool. You

can check that answer is boolean using: isinstance(answer, bool).

o The ask_question method must append the string '\nIs this statement true or false?' to

the end of the problem text.

o The get_answer method should return a string (to match the other

problems): str(self.answer)

o The check_answer method will receive a string, so we must compare a string version of

the correct answer to the received answer.

Because all problems are using the same method names, the quiz will be able to create a list of

Problems, and ask each problem to ask its question, check its answer, and tell us the correct

answer.

Exercise 20.3

Do the following (if you haven't already):

Add the FillInTheBlank and TrueFalse classes as described above.

Add some unit tests to the test cases at the end of the file(s).

Add at least one sample problem for each of the two new classes to the quiz.

Your quiz application should now operate as it did before, with randomly ordered questions of all

three subtypes.

Checking In

Submit the final version of all of your files. We will grade this exercise according to the following

criteria:

Correctness:

o 40% - exercise 13.2 - Add the required inheritance.

o 40% - exercise 13.3 - Configure the polymorphic behavior.

o 10% - New questions of each type are added

Understandability:

o 5% - Header Documentation - Document the basic purpose, authors and assignment number for each file.

o 5% - Code Documentation - Separate the logical blocks of your program with useful comments and white space.

Source Code:
Output: