

Assignment No. 1: Introduction to Java

Set A:

1. Write a java Program to check whether given number is Prime or Not.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int num = 29;  
        boolean flag = false;  
        for (int i = 2; i <= num / 2; ++i) {  
            // condition for nonprime number  
            if (num % i == 0) {  
                flag = true;  
                break;  
            }  
        }  
  
        if (!flag)  
            System.out.println(num + " is a prime number.");  
        else  
            System.out.println(num + " is not a prime number.");  
    }  
}
```

Output :

29 is a prime number

2. Write a java Program to display all the perfect numbers between 1 to n.

```
import java.util.Scanner;  
  
public class Perfect  
{  
  
    static boolean perfect(int num)  
    {  
  
        int sum = 0;
```

```

    for(int i=1; i<num; i++)
    {
        if(num%i==0)
        {
            sum = sum+i;
        }
    }
    if(sum==num)
        return true;
    else
        return false;
}

public static void main(String[] args)
{
    Scanner obj = new Scanner (System.in);
    System.out.println("enter the value for n");
    int n = obj.nextInt();
    for(int i=1; i<=n; i++)
    {
        if(perfect(i))
            System.out.println(i);
    }
}

}

```

Output :

enter the value for n

50

6

28

3. Write a java Program to accept employee name from a user and display it in reverse order.

```
import java.util.Scanner; // import the Scanner class
```

```
public class ReverseName {
```

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

```
        Scanner input = new Scanner(System.in); // create a Scanner object to read  
        input from the user
```

```
        System.out.print("Enter the employee's name: ");
```

```
        String name = input.nextLine(); // read the employee's name as a single  
        line of input
```

```
        // Reverse the name by printing each character in the name in reverse  
        order
```

```
        for (int i = name.length() - 1; i >= 0; i--) {
```

```
            System.out.print(name.charAt(i));
```

```
        }
```

```
        System.out.println(); // move to the next line after printing the reversed  
        name
```

```
        input.close(); // close the Scanner object to free up resources
```

```
    }
```

```
}
```

4. Write a java program to display all the even numbers from an array. (Use Command Line arguments)

```
public class OddEvenInArrayExample{  
    public static void main(String args[]){  
        int a[]={1,2,5,6,3,2};  
        System.out.println("Odd Numbers:");  
        for(int i=0;i<a.length;i++){  
            if(a[i]%2!=0){  
                System.out.println(a[i]);  
            }  
        }  
        System.out.println("Even Numbers:");  
        for(int i=0;i<a.length;i++){  
            if(a[i]%2==0){  
                System.out.println(a[i]);  
            }  
        }  
    }  
}
```

OUTPUT :

Odd Numbers:

1

5

3

Even Numbers:

2

6

5. Write a java program to display the vowels from a given string.

```
package cf.java.string;
```

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

```
public class JavaProgramToPrintVowelsInAString {
```

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

```
        // Declare a variables
```

```
        String str = null;
```

```
        Scanner sc = new Scanner(System.in);
```

```
        // Accept any string from user
```

```
        System.out.print("Enter any String: ");
```

```
        str = sc.nextLine();
```

```
        // Convert String to lower case letter
```

```
        str = str.toLowerCase();
```

```
        // Print all vowels
```

```
        System.out.print("Vowels in the given String are:");
```

```
        // For loop to iterate String
```

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

```
            // Check a character is vowel or not
```

```

        if (str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i)
== 'i' || str.charAt(i) == 'o'
        || str.charAt(i) == 'u') {
            // Display each vowel
            System.out.print(" " + str.charAt(i));
        }
    }
}

```

OUTPUT

Enter any String: Codingface

Vowels in the given String are: o i a e

Set B:

1. Write a java program to accept n city names and display them in ascending order.

```

class SortCity
{
    public static void main(String arg[])
    {
        String name[]=new String[10];
        int l=arg.length;
        String temp;
        for(int i=0;i<l;i++)
        {
            name[i]=arg[i];
        }
        for(int j=0;j<l;j++)
        {

```

```

        for(int k=j+1;k<l;k++)
        {
            if((name[j].compareTo(name[k]))<0)
            {
                temp=name[j];
                name[j]=name[k];
                name[k]=temp;
            }
        }
    }
    System.out.println("Sorted City Are-");
    for(int i=0;i<l;i++)
    {
        System.out.println(name[i]);
    }
}
}

```

/*OUTPUT

D:\javapro>javac SortCity.java

D:\javapro>java SortCity pune nagar solapurthane satara baramati

Sorted City Are-

thane

solapur

satara

pune

nagar

baramati

- 2. Write a java program to accept n numbers from a user store only Armstrong numbers in an array and display it.**

```
public class Armstrong {
```

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

```
        int number = 1634, originalNumber, remainder, result = 0, n = 0;
```

```

originalNumber = number;

for (;originalNumber != 0; originalNumber /= 10, ++n);

originalNumber = number;

for (;originalNumber != 0; originalNumber /= 10)
{
    remainder = originalNumber % 10;
    result += Math.pow(remainder, n);
}

if(result == number)
    System.out.println(number + " is an Armstrong number.");
else
    System.out.println(number + " is not an Armstrong number.");
}
}

```

OUTPUT

1634 is an Armstrong number.

3. Write a java program to search given name into the array, if it is found then display its index otherwise display appropriate message.

```
import java.util.Scanner; // import the Scanner class
```

```

public class SearchName {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in); // create a Scanner object to read
        input from the user

```

```

        // create an array of names

```

```

        String[] names = {"Alice", "Bob", "Charlie", "David", "Eve"};

```



```

System.out.print("Enter the name to search for: ");

String name = input.nextLine(); // read the name to search for as a single
line of input

// search for the name in the array
boolean found = false; // flag to track whether the name was found
for (int i = 0; i < names.length; i++) {
    if (names[i].equals(name)) {
        // name was found at index i
        found = true;
        System.out.println("Name found at index " + i);
        break;
    }
}

if (!found) {
    // name was not found
    System.out.println("Name not found in the array.");
}

input.close(); // close the Scanner object to free up resources
}
}

```

4. Write a java program to display following pattern:

5

4 5

3 4 5

2 3 4 5

1 2 3 4 5

```
public class Pattern {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 5; i++) {  
            for (int j = i; j <= 5; j++) {  
                System.out.print(j);  
            }  
            System.out.println();  
        }  
    }  
}
```

5. Write a java program to display following pattern:

1

0 1

0 1 0

1 0 1 0

```
public class Pattern {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 4; i++) {  
            for (int j = 1; j <= i; j++) {  
                if ((i + j) % 2 == 0) {  
                    System.out.print("0");  
                } else {  
                    System.out.print("1");  
                }  
            }  
        }  
    }  
}
```

```

    }
}
System.out.println();
}
}
}
}

```

Set C:

1. Write a java program to count the frequency of each character in a given string.

```

public class Main
{
    public static void main(String[] args)
    {
        String str = "Study Tonight";
        int[] freq = new int[str.length()];
        System.out.println("The entered string is "+str);
        //Convert the given string into character array
        char str1[] = str.toCharArray();
        for(int i = 0; i < str.length(); i++)
        {
            freq[i] = 1;
            for(int j = i+1; j < str.length(); j++)
            {
                if(str1[i] == str1[j])
                {
                    freq[i]++;

                    //Set str1[j] to 0 to avoid printing visited character
                    str1[j] = '0';
                }
            }
        }
        //Displays the characters and their corresponding frequency
    }
}

```

```
System.out.println("Frequencies of the characters in the string are  
as below: ");
```

```
System.out.println("Characters frequencies");  
for(int i = 0; i < freq.length; i++)  
{  
    if(str1[i] != ' ' && str1[i] != '0')  
        System.out.println(str1[i] + "      " + freq[i]);  
}  
}
```

OUTPUT :

The entered string is Study Tonight

Frequencies of the characters in the string are as below:

Characters frequencies

S 1

t 2

u 1

d 1

y 1

T 1

o 1

n 1

i 1

g 1

h 1

2. Write a java program to display each word in reverse order from a string array.

```
public class Example
```

```
{
```

```
    public void reverseWordInMyString(String str)
```

```
    {
```

```
        /* The split() method of String class splits  
        * a string in several strings based on the  
        * delimiter passed as an argument to it  
        */
```

```
        String[] words = str.split(" ");
```

```
        String reversedString = "";
```

```

        for (int i = 0; i < words.length; i++)
        {
            String word = words[i];
            String reverseWord = "";
            for (int j = word.length()-1; j >= 0; j--)
            {
                /* The charAt() function returns the character
                * at the given position in a string
                */
                reverseWord = reverseWord + word.charAt(j);
            }
            reversedString = reversedString + reverseWord + " ";
        }
        System.out.println(str);
        System.out.println(reversedString);
    }
    public static void main(String[] args)
    {
        Example obj = new Example();
        obj.reverseWordInMyString("Welcome to BeginnersBook");
        obj.reverseWordInMyString("This is an easy Java Program");
    }
}

```

OUTPUT:

```

Welcome to BeginnersBook
emocleW ot kooBsrennigeB
This is an easy Java Program
sihT si na ysae avaJ margorP

```

3. Write a java program for union of two integer array.

```

import java.util.Arrays;
import java.util.HashSet;

public class Main
{
    public static void main(String[] args)
    {
        Integer[] firstArray = {0,2,4,6,8};
    }
}

```

```

Integer[] secondArray = {1,3,5,7,9};

HashSet<Integer> set = new HashSet<>();

set.addAll(Arrays.asList(firstArray));

set.addAll(Arrays.asList(secondArray));

System.out.println(set);

//convert to array
Integer[] union = {};
union = set.toArray(union);

System.out.println(Arrays.toString(union));
}
}

```

OUTPUT:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

4. Write a java program to display transpose of given matrix.

```

import java.util.Scanner;

public class Transpose
{
    public static void main(String args[])
    {
        int i, j;
        System.out.println("Enter total rows and columns: ");
        Scanner s = new Scanner(System.in);
        int row = s.nextInt();
        int column = s.nextInt();
        int array[][] = new int[row][column];
        System.out.println("Enter matrix:");
        for(i = 0; i < row; i++)
        {
            for(j = 0; j < column; j++)
            {

```

```

        array[i][j] = s.nextInt();
        System.out.print(" ");
    }
}
System.out.println("The above matrix before Transpose is ");
for(i = 0; i < row; i++)
{
    for(j = 0; j < column; j++)
    {
        System.out.print(array[i][j]+" ");
    }
    System.out.println(" ");
}
System.out.println("The above matrix after Transpose is ");
for(i = 0; i < column; i++)
{
    for(j = 0; j < row; j++)
    {
        System.out.print(array[j][i]+" ");
    }
    System.out.println(" ");
}
}
}

```

OUTPUT:

\$ javac Transpose.java

\$ java Transpose

Enter total rows and columns:

3 3

Enter matrix:

1

2

3

4

5

6

7

8

9

The above matrix before Transpose is

1 2 3

4 5 6

7 8 9

The above matrix after Transpose is

1 4 7

2 5 8

3 6 9

5. Write a java program to display alternate character from a given string.

```
import java.util.Scanner; // import the Scanner class
```

```
public class AlternateCharacters {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in); // create a Scanner object  
        to read input from the user
```

```
        System.out.print("Enter a string: ");  
        String str = input.nextLine(); // read the string as a single line of  
        input
```

```
        // print the alternate characters of the string  
        for (int i = 0; i < str.length(); i += 2) {  
            System.out.print(str.charAt(i));  
        }  
        System.out.println(); // move to the next line after printing the  
        alternate characters
```

```
        input.close(); // close the Scanner object to free up resources  
    }  
}
```


ASSIGNMENT 2 : CLASSES, OBJECTS AND METHODS

Set A:

1. Write a Java program to calculate power of a number using recursion.

```
class Power {  
    public static void main(String[] args) {  
  
        int base = 3, powerRaised = 4;  
        int result = power(base, powerRaised);  
  
        System.out.println(base + "^" + powerRaised + "=" + result);  
    }  
  
    public static int power(int base, int powerRaised) {  
        if (powerRaised != 0) {  
  
            // recursive call to power()  
            return (base * power(base, powerRaised - 1));  
        }  
        else {  
            return 1;  
        }  
    }  
}
```

OUTPUT:

3^4 = 81

3 * 3 * 3 * 3 = 81

2. Write a Java program to display Fibonacci series using function.

```
class Main {  
    public static void main(String[] args) {  
  
        int n = 10, firstTerm = 0, secondTerm = 1;  
        System.out.println("Fibonacci Series till " + n + " terms:");  
  
        for (int i = 1; i <= n; ++i) {  
            System.out.print(firstTerm + ", ");
```

```

        // compute the next term
        int nextTerm = firstTerm + secondTerm;
        firstTerm = secondTerm;
        secondTerm = nextTerm;
    }
}

```

OUTPUT :

Fibonacci Series till 10 terms:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34

- 3. Write a Java program to calculate area of Circle, Triangle & Rectangle.(Use Method Overloading)**

```

class OverloadDemo
{
    void area(float x)
    {
        System.out.println("the area of the square is "+Math.pow(x, 2)+"
sq units");
    }
    void area(float x, float y)
    {
        System.out.println("the area of the rectangle is "+x*y+" sq units");
    }
    void area(double x)
    {
        double z = 3.14 * x * x;
        System.out.println("the area of the circle is "+z+" sq units");
    }
}
class Overload
{
    public static void main(String args[])
    {
        OverloadDemo ob = new OverloadDemo();
        ob.area(5);
        ob.area(11,12);
        ob.area(2.5);
    }
}

```

```
}  
}
```

OUTPUT :

the area of the square is 25.0 sq units

the area of the rectangle is 132.0 sq units

the area of the circle is 19.625 sq units

4. Write a Java program to Copy data of one object to another Object.

```
public class CopyValuesConstructor {  
    int id;  
    String name;  
  
    CopyValuesConstructor(int id, String name){  
        System.out.println("Constructor called.");  
        this.id = id;  
        this.name = name;  
    }  
  
    //Copy value of one object into another.  
    CopyValuesConstructor(CopyValuesConstructor obj){  
        System.out.println("Constructor called for copying value.");  
        this.id = obj.id;  
        this.name = obj.name;  
    }  
  
    public static void main(String args[]){  
        //parameterized constructor call  
        CopyValuesConstructor obj1 = new CopyValuesConstructor(10,  
"W3spoint");  
  
        //print values of object properties.  
        System.out.println("obj1 id = " + obj1.id);  
        System.out.println("obj1 name = " + obj1.name);  
        //Constructor call to copy the value of one object into other.  
        CopyValuesConstructor obj2 = new  
CopyValuesConstructor(obj1);  
        //print values of object properties.  
        System.out.println("obj2 id = " + obj2.id);  
    }  
}
```

```

        System.out.println("obj2 name= " + obj2.name);
    }
}

```

OUTPUT :

Constructor called.

obj1 id = 10

obj1 name = W3spoint

Constructor called for copying value.

obj2 id = 10

obj2 name= W3spoint

5. Write a Java program to calculate factorial of a number using recursion.

```

public class Factorial {

    public static void main(String[] args) {
        int num = 6;
        long factorial = multiplyNumbers(num);
        System.out.println("Factorial of " + num + " = " + factorial);
    }

    public static long multiplyNumbers(int num)
    {
        if (num >= 1)
            return num * multiplyNumbers(num - 1);
        else
            return 1;
    }
}

```

OUTPUT :

Factorial of 6 = 720

6 * 5 * 4 * 3 * 2 * 1 * 1 (for 0) = 720

Set B:

1. Define a class person(pid,pname,age,gender). Define Default and parameterised constructor. Overload the constructor. Accept the 5 person details and display it.(use this keyword).

```
public class Person {  
    // instance variables  
    private int pid;  
    private String pname;  
    private int age;  
    private char gender;  
  
    // default constructor  
    public Person() {  
        pid = 0;  
        pname = "";  
        age = 0;  
        gender = ' ';  
    }  
  
    // parameterized constructor  
    public Person(int pid, String pname, int age, char gender) {  
        this.pid = pid;  
        this.pname = pname;  
        this.age = age;  
        this.gender = gender;  
    }  
  
    // overloaded constructor  
    public Person(Person p) {  
        this(p.pid, p.pname, p.age, p.gender);  
    }  
  
    // getter and setter methods for the instance variables  
    public int getPid() {  
        return pid;  
    }  
}
```

```
public void setPid(int pid) {  
    this.pid = pid;  
}
```

```
public String getPName() {  
    return pname;  
}
```

```
public void setPname(String pname) {  
    this.pname = pname;  
}
```

```
public int getAge() {  
    return age;  
}
```

```
public void setAge(int age) {  
    this.age = age;  
}
```

```
public char getGender() {  
    return gender;  
}
```

```
public void setGender(char gender) {  
    this.gender = gender;  
}
```

// toString method to display the person's details

@Override

```
public String toString() {  
    return "Person{" +  
        "pid=" + pid +  
        ", pname=" + pname + "\" +  
        ", age=" + age +  
        ", gender=" + gender +  
        '}';  
}
```

```
}  
}
```

```
import java.util.Scanner; // import the Scanner class
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in); // create a Scanner object  
        to read input from the user
```

```
        // create an array of Person objects  
        Person[] people = new Person[5];
```

```
        // accept the details of 5 people  
        for (int i = 0; i < 5; i++) {  
            System.out.println("Enter the details of person " + (i + 1) + ":");  
            System.out.print("Enter the ID: ");  
            int id = input.nextInt();  
            input.nextLine(); // consume the remaining newline character  
            System.out.print("Enter the name: ");  
            String name = input.nextLine();  
            System.out.print("Enter the age: ");  
            int age = input.nextInt();  
            input.nextLine(); // consume the remaining newline character
```

3. Define a class product(pid,pname,price). Write a function to accept the product details, to display product details and to calculate total amount. (use array of Objects)

```
public class Product {  
    // instance variables  
    private int pid;  
    private String pname;  
    private double price;  
  
    // default constructor  
    public Product() {  
        pid = 0;  
        pname = "";  
        price = 0.0;  
    }  
  
    // parameterized constructor  
    public Product(int pid, String pname, double price) {  
        this.pid = pid;  
        this.pname = pname;  
        this.price = price;  
    }  
  
    // getter and setter methods for the instance variables  
    public int getPid() {  
        return pid;  
    }  
}
```



```
public void setPid(int pid) {  
    this.pid = pid;  
}
```

```
public String getPname() {  
    return pname;  
}
```

```
public void setPname(String pname) {  
    this.pname = pname;  
}
```

```
public double getPrice() {  
    return price;  
}
```

```
public void setPrice(double price) {  
    this.price = price;  
}
```

// toString method to display the product's details

@Override

```
public String toString() {  
    return "Product{" +  
        "pid=" + pid +
```

```
        ", pname=" + pname + "\" +  
        ", price=" + price +  
        '}'  
    }  
}
```

```
import java.util.Scanner; // import the Scanner class
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in); // create a Scanner object to read  
        input from the user
```

```
        // create an array of Product objects
```

```
        Product[] products = new Product[5];
```

```
        // accept the details of 5 products
```

```
        for (int i = 0; i < 5; i++) {
```

```
            System.out.println("Enter the details of product " + (i + 1) + ":");
```

```
            System.out.print("Enter the ID: ");
```

```
            int id = input.nextInt();
```

```
            input.nextLine(); // consume the remaining newline character
```

```
            System.out.print("Enter the name: ");
```

```
            String name = input.nextLine();
```

```
            System.out.print("Enter the price: ");
```

```
            double price = input.nextDouble();
```

```
            input.nextLine(); // consume the remaining newline character
```

```

// create a Product object and set its details
Product p = new Product(id, name, price);
products[i] = p;
}

// display the details of the products
for (Product p : products) {
    System.out.println(p);
}

// calculate the total amount
double totalAmount = 0.0;
for (Product p : products) {
    totalAmount += p.getPrice();
}
System.out.println("Total amount: " + totalAmount);

input.close(); // close the Scanner object to free up resources
}
}

```

3. Define a class Student(rollno,name,per). Create n objects of the student class and Display it using toString().(Use parameterized constructor)

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

```
class student
```

```
{

    int rno;

    String name;

    float per;

    student()

    {

        System.out.println("You are in default constructor :");

    }

    student(int rno,String name,float per)

    {

        this.rno=rno;

        this.name=name;

        this.per=per;

    }

    static int cnt=0;

    static void obj()

    {

        cnt++;

    }

}
```

```
        System.out.println("Object created :"+cnt);
    }

    void display()
    {
        System.out.println("Student roll no:"+rno);

        System.out.println("Student name:"+name);

        System.out.println("Student percentage:"+per);
    }

    public static void main(String a[])
    {
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter How many student information
do you want to store ?");

        int n=sc.nextInt();

        student s=new student();

        student s1[]=new student[n];

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

            System.out.println("Enter Student roll no :");
```

```

        int rno=sc.nextInt();

        System.out.println("Enter Student name :");

        String name=sc.next();

        System.out.println("Enter Student percentage :");

        float per=sc.nextFloat();

        s1[i]=new student(rno,name,per);

        s1[i].obj();

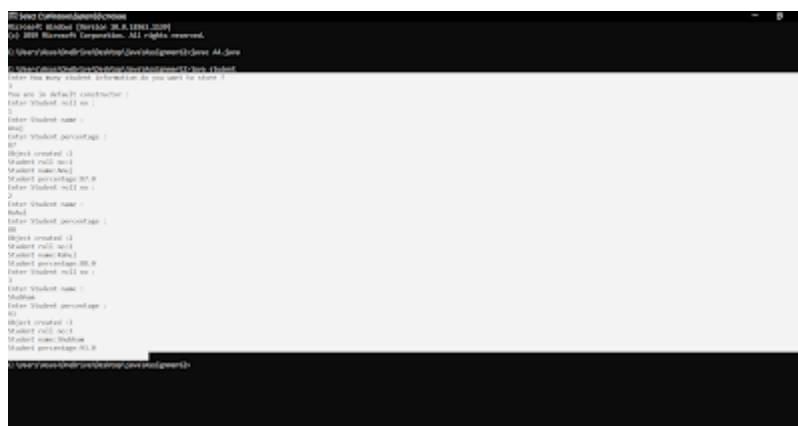
        s1[i].display();

    }

}
}

```

OUTPUT :



```

C:\Users\Chaitanya> javac Student.java
C:\Users\Chaitanya> java Student
Enter Student roll no :
1
Enter Student name :
Anil
Enter Student percentage :
87
Object created :1
Student roll no:1
Student name:Anil
Student percentage:87.0
Enter Student roll no :
2
Enter Student name :
Rahul
Enter Student percentage :
88
Object created :2
Student roll no:2
Student name:Rahul
Student percentage:88.0
Enter Student roll no :
3
Enter Student name :
Shubham
Enter Student percentage :
89
Object created :3
Student roll no:3
Student name:Shubham
Student percentage:89.0

```

4. Define a class MyNumber having one private integer data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value. Write methods is Negative, is Positive. Use command line argument to pass a value to the object and perform the

above tests.

```
public class MyNumber {  
    private int x;  
    public MyNumber(){  
        x=0;  
    }  
    public MyNumber(int x){  
        this.x=x;  
    }  
    public boolean isNegative(){  
        if(x<0)  
            return true;  
        else return false;  
    }  
    public boolean isPositive(){  
        if(x>0)  
            return true;  
        else return false;  
    }  
    public boolean isZero(){  
        if(x==0)  
            return true;  
        else return false;  
    }  
    public boolean isOdd(){  
        if(x%2!=0)  
            return true;  
        else return false;  
    }  
    public boolean isEven(){  
        if(x%2==0)  
            return true;  
        else return false;  
    }  
  
    public static void main(String [] args) throws  
        ArrayIndexOutOfBoundsException
```

```

{
    int x=Integer.parseInt(args[0]);
    MyNumber m=new MyNumber(x);
    if(m.isNegative())
        System.out.println("Number is Negative");
    if(m.isPositive())
        System.out.println("Number is Positive");
    if(m.isEven())
        System.out.println("Number is Even");
    if(m.isOdd())
        System.out.println("Number is Odd");
    if(m.isZero())
        System.out.println("Number is Zero");
}

}

```

Output:

Number is Negative

Number is Odd

Set C:

1. Define class Student(rno, name, mark1, mark2). Define Result class(total, percentage) inside the student class. Accept the student details & display the mark sheet with rno,name, mark1, mark2, total, percentage. (Use inner class concept)
import java.util.*;

class student

```
{
```

```
    int rno;
```


String name;

int marks1,marks2;

Scanner sc=new Scanner(System.in);

class result

{

int total;

double per;

void accept()

{

System.out.println("Enter Student Roll no:");

rno=sc.nextInt();

System.out.println("Enter Student Name:");

name=sc.next();

System.out.println("Enter Marks1:");

marks1=sc.nextInt();

System.out.println("Enter Marks2:");

marks2=sc.nextInt();

}

void display()

```

    {

        System.out.println("Student Roll no :"+rno);

        System.out.println("Student Name:"+name);

        System.out.println("Student Marks1:"+marks1);

        System.out.println("Student Marks2:"+marks2);

        total=marks1+marks2;

        System.out.println("Total marks:"+total);

        per=total/2.0;

        System.out.println("Total percentage:"+per);

    }

}

public static void main(String a[])

{

    student s=new student();

    student.result sr=s.new result();

    sr.accept();

    sr.display();

}

```

```

}

C:\Users\Asus\OneDrive\Desktop\java\Assignment2>javac BS.java
C:\Users\Asus\OneDrive\Desktop\java\Assignment2>java student
Enter Student Roll no:
1
Enter Student Name:
shubham
Enter Marks1:
87
Enter Marks2:
89
Student Roll no :1
Student Name:shubham
Student Marks1:87
Student Marks2:89
Total marks:176
Total percentage:88.0
C:\Users\Asus\OneDrive\Desktop\java\Assignment2>

```

2. Write a java program to accept n employee names from user. Sort them in ascending order and Display them.(Use array of object and Static keyword)

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

```
class SortStrings
```

```

{
    public static void main(String args[])
    {
        String temp;
        Scanner SC = new Scanner(System.in);

        System.out.print("Enter the value of N: ");
        int N= SC.nextInt();
        SC.nextLine(); //ignore next line character

        String names[] = new String[N];

        System.out.println("Enter names: ");
        for(int i=0; i<N; i++)
        {
            System.out.print("Enter name [ " + (i+1) + " ]: ");
            names[i] = SC.nextLine();
        }

        //sorting strings

        for(int i=0; i<5; i++)

```

```

{
    for(int j=1; j<5; j++)
    {
        if(names[j-1].compareTo(names[j])>0)
        {
            temp=names[j-1];
            names[j-1]=names[j];
            names[j]=temp;
        }
    }
}

```

```

        System.out.println("\nSorted names are in Ascending Order: ");
        for(int i=0;i<N;i++)
        {
            System.out.println(names[i]);
        }
    }
}

```

OUTPUT:

Enter the value of N: 5

Enter names:

Enter name [1]: Mike

Enter name [2]: Alex

Enter name [3]: Zoya

Enter name [4]: Yashin

Enter name [5]: Bobby

Sorted names are in Ascending Order:

Alex

Bobby

Mike

Yashin

Zoya

3. Write a java program to accept details of 'n' cricket players(pid, pname, totalRuns,InningsPlayed, NotOuttimes). Calculate the average of all the players. Display the details of player having maximum average.

```
import java.io.*;
class Player
{
    String name;
    int TotalRuns,TimesNotOut,InningsPlayed;
    float avg;
    static BufferedReader br=new BufferedReader(new
    InputStreamReader(System.in));
    void getData()
    {
        try
        {
            System.out.println("Enter player name:");
            name=br.readLine();
            System.out.println("Enter Total Runs:");
            TotalRuns=Integer.parseInt(br.readLine());
            System.out.println("Enter times Not Out:");
            TimesNotOut=Integer.parseInt(br.readLine());
            System.out.println("Enter the Innings Played:");
            InningsPlayed=Integer.parseInt(br.readLine());
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    void putdata()
    {
        System.out.println(name+"\t\t"+TotalRuns+"\t\t"+TimesNotOut+
        "\t\t"+InningsPlayed+"\t\t"+avg);
    }
    float getAvg()
    {
```

```

    avg=(TotalRuns/(InningsPlayed-TimesNotOut+1));
    return avg;
}
}
public class Main
{
    static BufferedReader br=new BufferedReader(new
    InputStreamReader(System.in));
    public static void main(String args[])
    {
        try
        {
            System.out.println("Enter the Number of Players:");
            int n=Integer.parseInt(br.readLine());
            Player p[]=new Player[n];
            for(int i=0;i<n;i++)
            {
                p[i]=new Player();
                p[i].getData();
            }
            sortPlayer(p);
            System.out.println("Name"+"\\t"+"TotalRuns"+"\\t"+"TimesNotOut"
            +"\\t"+"InningsPlayed"+"\\t"+"Average");
            for(int i=0;i<n;i++)
            {
                p[i].putdata();
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public static void sortPlayer(Player p[])
    {
        int n=p.length;
        for(int i=0;i<n;i++)

```

```

{
for(int j=i+1;j<n;j++)
{
if(p[i].getAvg()<p[j].getAvg())
{
Player t=p[i];
p[i]=p[j];
p[j]=t;
}
}
}
}
}
}

```

4. Write a java program to accept details of 'n' books. And Display the quantity of given book.

```

import java.util.Scanner; // import the Scanner class
import java.util.HashMap; // import the HashMap class

```

```

public class Books {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in); // create a Scanner
        object to read input from the user

```

```

        // create a HashMap to store the books
        HashMap<String, Integer> books = new HashMap<>();

```

```

        System.out.print("Enter the number of books: ");
        int n = input.nextInt();
        input.nextLine(); // consume the remaining newline character

```

```

        // accept the details of the books
        for (int i = 0; i < n; i++) {
            System.out.print("Enter the name of book " + (i + 1) + ": ");
            String name = input.nextLine();
            System.out.print("Enter the quantity of book " + (i + 1) + ": ");
            int quantity = input.nextInt();

```

```
        input.nextLine(); // consume the remaining newline character

        // add the book and its quantity to the HashMap
        books.put(name, quantity);
    }

    System.out.print("Enter the name of the book to search for: ");
    String name = input.nextLine();

    // check if the book is in the HashMap
    if (books.containsKey(name)) {
        // book is in the HashMap
        System.out.println("Quantity of book '" + name + "': " +
books.get(name));
    } else {
        // book is not in the HashMap
        System.out.println("Book '" + name + "' not found.");
    }

    input.close(); // close the Scanner object to free up resources
}
}
```


Assignment No. 3: Inheritance, Package and Collection

Set A:

1. Write a java program to calculate area of Cylinder and Circle.(Use super keyword)

```
public class Shape {  
    // instance variables  
    private double radius;  
  
    // default constructor  
    public Shape() {  
        radius = 0.0;  
    }  
  
    // parameterized constructor  
    public Shape(double radius) {  
        this.radius = radius;  
    }  
  
    // getter and setter methods for the radius  
    public double getRadius() {  
        return radius;  
    }  
  
    public void setRadius(double radius) {  
        this.radius = radius;  
    }  
}
```

```
public class Circle extends Shape {  
    // instance variables  
    private final double PI = 3.14;  
  
    // default constructor  
    public Circle() {  
        super();  
    }  
  
    // parameterized constructor  
    public Circle(double radius) {  
        super(radius);  
    }  
  
    // method to calculate the area of the circle  
    public double calcArea() {  
        return PI * getRadius() * getRadius();  
    }  
}
```

```
public class Cylinder extends Circle {  
    // instance variables  
    private double height;  
  
    // default constructor
```

```
public Cylinder() {  
    super();  
    height = 0.0;  
}
```

// parameterized constructor

```
public Cylinder(double radius, double height) {  
    super(radius);  
    this.height = height;  
}
```

// getter and setter methods for the height

```
public double getHeight() {
```

2. Define an Interface Shape with abstract method area(). Write a java program to calculate an area of Circle and Sphere.(use final keyword)

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

```
interface Shape
```

```
{
```

```
    final float pi=3.14f;
```

```
    float area(int r);
```

```
}
```

```
class Circle implements Shape
```

```
{
```

```
    public float area(int r)
```

```
    {
```

```
        return(pi*r*r);
```

```
}
```

```
}
```

class Sphere implements Shape

```
{
```

```
    public float area(int r)
```

```
    {
```

```
        return(4*pi*r*r);
```

```
    }
```

```
}
```

class ShapeDemo

```
{
```

```
    public static void main(String arg[])
```

```
    {
```

```
        int r;
```

```
        BufferedReader din=new BufferedReader(new  
InputStreamReader(System.in));
```

```
        try
```

```
        {
```

```
            System.out.print("Enter radius-");
```

```
            r=Integer.parseInt(din.readLine());
```

```
            Circle c1=new Circle();
```

```
            Sphere s1=new Sphere();
```

```
            System.out.println("Area of Circle:"+c1.area(r));
```

```
            System.out.println("Area of Sphere:"+s1.area(r));
```

```
        }
```

```
        catch(Exception e)
```

```
        {
```

```

    System.out.println(e.getMessage());
}
}
}
/*Output
D:\javapro\Ass2>javac ShapeDemo.java

```

```

D:\javapro\Ass2>java ShapeDemo

```

Enter radius-5

Area of Circle:78.5

Area of Sphere:314.0

3. Define an Interface "Integer" with a abstract method check(). Write a Java program to check whether a given number is Positive or Negative.

```

import java.util.Scanner;

public class PositiveOrNegative {
    public static void main(String args[]){
        int num;
        System.out.println("Enter a number ::");
        Scanner sc = new Scanner(System.in);
        num = sc.nextInt();

        if (num > 0){
            System.out.println("Given number is a positive integer");
        } else if(num < 0){
            System.out.println("Given number is a negative integer");
        } else {
            System.out.println("Given number is neither positive nor negative integer");
        }
    }
}

```

```
}
```

OUTPUT:

Enter a number ::

55

Given number is a positive integer

OUTPUT 2:

Enter a number ::

-88

Given number is a negative integer

4. Define a class Student with attributes roll no and name. Define default and parameterized constructor. Override the to String () method. Keep the count of Objects created. Create objects using parameterized constructor and display the object count after each object is created.

```
public class Student {
```

```
    // instance variables
```

```
    private int rollno;
```

```
    private String name;
```

```
    // static variable to keep track of the object count
```

```
    private static int count = 0;
```

```
    // default constructor
```

```
    public Student() {
```

```
        rollno = 0;
```

```
        name = "";
```

```
        count++;
```

```
    }
```

```
    // parameterized constructor
```

```
    public Student(int rollno, String name) {
```

```
    this.rollno = rollno;
    this.name = name;
    count++;
}
```

// getter and setter methods for the instance variables

```
public int getRollno() {
    return rollno;
}
```

```
public void setRollno(int rollno) {
    this.rollno = rollno;
}
```

```
public String getName() {
    return name;
}
```

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

// toString method to display the student's details

@Override

```
public String toString() {
    return "Student{" +
```

```
        "rollno=" + rollno +  
        ", name=" + name + "\" +  
        '}'  
    }  
}
```

```
// static method to get the object count  
public static int getCount() {  
    return count;  
}  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        // create Student objects using the parameterized constructor  
        Student s1 = new Student(1, "Alice");  
        System.out.println(s1);  
        System.out.println("Number of Student objects created: " +  
Student.getCount());  
    }  
}
```

5. Write a java program to accept 'n' integers from the user & store them in an Array List collection. Display the elements of Array List collection in reverse order.

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

```
class array
```

```
{
```



```
public static void main(String a[])

{

    Scanner sc=new Scanner(System.in);

    System.out.println("Enter Limit of ArrayList :");

    int n=sc.nextInt();

    ArrayList alist=new ArrayList();

    System.out.println("Enter Elements of ArrayList :");

    for(int i=0;i<n;i++)

    {

        String elmt=sc.next();

        alist.add(elmt);

    }

    System.out.println("Original ArrayList is :"+alist);
```

```
Collections.reverse(alist);
```

```
System.out.println("Reverse of a ArrayList is :"+alist);
```

```
}
```

```
}
```

Output :-

```
C:\Windows\System32\cmd.exe
C:\Users\Asus\OneDrive\Desktop\java\Slips>java array
Enter Limit of ArrayList :
6
Enter Elements of ArrayList :
43
26
87
56
97
12
Original ArrayList is :[43, 26, 87, 56, 97, 12]
Reverse of a ArrayList is :[12, 97, 56, 87, 26, 43]
C:\Users\Asus\OneDrive\Desktop\java\Slips>
```

Set B:

1. Create an abstract class Shape with methods `calc_area()` & `calc_volume()`. Derive two classes `Sphere(radius)` & `Cone(radius, height)` from it. Calculate area and volume of both. (Use Method Overriding)
abstract class shape

```
{
```

```
    abstract void area();

    abstract void volume();

}

class sphere extends shape
{

    double pi=3.14;

    double radius=4.35;

    void area()

    {

        double ar=4*pi*radius*radius;

        System.out.println("Area of Sphere is :"+ar);

    }

    void volume()

    {

        double vol=(4/3)*(pi*radius*radius*radius);

        System.out.println("Volume of Sphere is :"+vol);

    }

}

class cone extends shape
```

```
{

    double pi=3.14;

    double radius=4.35;

    double height=6.45;

    void area()

    {

        double tot=pi*radius*height;

        System.out.println("Area of cone is :"+tot);

    }

    void volume()

    {

        double vol=(pi*radius*radius)*(height/3);

        System.out.println("Volume of Cone is :"+vol);

    }

}

class cylinder extends shape

{

    double pi=3.14;
```

```
double radius=4.35;
```

```
double height=6.45;
```

```
void area()
```

```
{
```

```
    double A=(2*pi*radius*height)+(2*pi*radius*radius);
```

```
    System.out.println("Area of Cylinder is :"+A);
```

```
}
```

```
void volume()
```

```
{
```

```
    double V=pi*radius*radius*height;
```

```
    System.out.println("Volume of Cylinder is :"+V);
```

```
}
```

```
}
```

```
class box extends shape
```

```
{
```

```
    int length=5;
```

```
    int breadth=7;
```

```
    double height=4.35;
```

```
    void area()
```

```
{

    double
a=(2*height*breadth)+(2*height*length)+(2*breadth*length);

    System.out.println("Area of Box is :"+a);

}

void volume()

{

    double v=length*breadth*height;

    System.out.println("Volume of Box is :"+v);

}

public static void main(String a[])

{

    sphere s=new sphere();

    s.area();

    s.volume();

    cone c=new cone();

    c.area();

    c.volume();

    cylinder cy=new cylinder();
```

```

        cy.area();

        cy.volume();

        box b=new box();

        b.area();

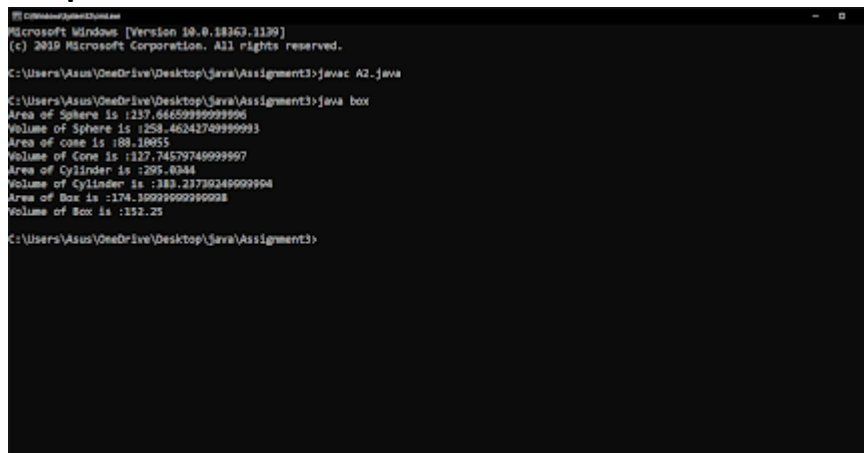
        b.volume();

    }

}

```

Output :-



```

C:\Users\Asus\OneDrive\Desktop\Java\Assignment3>javac A2.java

C:\Users\Asus\OneDrive\Desktop\Java\Assignment3>java box
Area of Sphere is :1257.6650999999994
Volume of Sphere is :1258.46241749999993
Area of cone is :88.18055
Volume of Cone is :127.74579749999997
Area of Cylinder is :295.8344
Volume of Cylinder is :388.23738249999994
Area of Box is :174.19099999999998
Volume of Box is :152.25

C:\Users\Asus\OneDrive\Desktop\Java\Assignment3>

```

2. Define a class Employee having private members-id, name, department, salary. Define default & parameterized constructors. Create a subclass called Manager with private member bonus. Define methods accept & display in both the classes. Create n objects of the manager class & display the details of the manager having the maximum total salary(salary+bonus).

```

import java.io.*;
class Employee
{
    private int id;

```

```

private String name, department;
private float salary;

Employee()
{
    id = 1;
    name = "nived";
    department = "bcs";
    salary = 20000;
}

Employee(int id, String name, String department, float salary)
{
    this.id = id;
    this.name = name;
    this.department = department;
    this.salary = salary;
}

void accept() throws IOException
{
    BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

    System.out.println("Enter employee information: id,
name, department, salary");

    id = Integer.parseInt(br.readLine());
    name= br.readLine();
    department = br.readLine();
    salary = Float.parseFloat(br.readLine());
}

void display()
{
    System.out.println("\nId : "+id+"\nName :
"+name+"\nDepartment : "+department+"\nSalary : "+salary);

```



```

    }

    float getsalary()
    {
        return salary;
    }
}

class Manager extends Employee
{
    private float bonus;

    Manager()
    {
        super();
    }

    Manager(int id, String name, String department, float salary,
float bonus)
    {
        super(id, name, department, salary);
        this.bonus = bonus;
    }

    void accept() throws IOException
    {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

        super.accept();

        System.out.println("Enter bonus : ");
        bonus = Float.parseFloat(br.readLine());
    }

    void display()
    {

```

```

        super.display();
        System.out.println("\nBonus : "+bonus);
    }

    float getbonus()
    {
        return bonus;
    }
}

```

3. Construct a Linked List containing name: CPP, Java, Python and PHP. Then extend your program to do the following:

- i. Display the contents of the List using an iterator**
- ii. Display the contents of the List in reverse order using a ListIterator.**

```
import java.util.LinkedList;
```

```
import java.util.ListIterator;
```

```

public class LinkedListExample {
    public static void main(String[] args) {
        // Create a linked list and add elements to it
        LinkedList<String> linkedList = new LinkedList<>();
        linkedList.add("CPP");
        linkedList.add("Java");
        linkedList.add("Python");
        linkedList.add("PHP");
        // Display the contents of the list using an iterator
        System.out.println("Displaying the contents of the list using an iterator:");
        ListIterator<String> iterator = linkedList.listIterator();
        while (iterator.hasNext()) {
            System.out.println(iterator.next());
        }
    }
}

```

```
// Display the contents of the list in reverse order using a list iterator
System.out.println("\nDisplaying the contents of the list in reverse order
using a list iterator:");

ListIterator<String> reverseliterator = linkedList.listIterator(linkedList.size());
while (reverseliterator.hasPrevious()) {
    System.out.println(reverseliterator.previous());
}
}
}
```

This code will output the following:

Displaying the contents of the list using an iterator:

CPP

Java

Python

PHP

Displaying the contents of the list in reverse order using a list iterator:

PHP

Python

Java

CPP

4. Create a hashtable containing employee name & salary. Display the details of the hashtable. Also search for a specific Employee and display salary of that employee.

```
import java.util.Hashtable;

public class EmployeeHashtable {
    public static void main(String[] args) {
        // Create a hashtable to store employee names and salaries
        Hashtable<String, Double> employeeTable = new Hashtable<>();

        // Add some employees and their salaries to the hashtable
        employeeTable.put("John", 45000.00);
        employeeTable.put("Jane", 50000.00);
        employeeTable.put("Bob", 55000.00);
        employeeTable.put("Alice", 60000.00);

        // Display the contents of the hashtable
        System.out.println("Employee salaries:");
        for (String name : employeeTable.keySet()) {
            System.out.println(name + ": $" + employeeTable.get(name));
        }

        // Search for a specific employee and display their salary
        String searchName = "Jane";
        if (employeeTable.containsKey(searchName)) {
            System.out.println(searchName + "'s salary is $" +
employeeTable.get(searchName));
        } else {
            System.out.println("Employee not found");
        }
    }
}
```

Output:

Employee salaries:

John: \$45000.0

Jane: \$50000.0

Bob: \$55000.0

Alice: \$60000.0

Jane's salary is \$50000.0

- 5. Write a package game which will have 2 classes Indoor & Outdoor.
Use a function display() to generate the list of player for the specific
game. Use default & parameterized constructor.**

```
package games;
public class Indoor
{
    protected String player;
    public Indoor()    {    }
    public Indoor(String p)
    {    player = p;    }
    public void display()
    {
        System.out.println(player);
    }

    protected void finalize()
    {
        System.out.println("Terminating Indoor...");
    }
}
```

Outdoor.java

```
package games;
public class Outdoor
{
    protected String player;
    public Outdoor(){    }
    public Outdoor(String p)
    {player = p; }
    public void display()
```

```
{ System.out.println(player); }  
protected void finalize()  
{System.out.println("Terminating Outdoor..."); }  
}
```

TestGamesPackage.java

```
import games.*;  
public class TestGamesPackage  
{  
    public static void main(String args[])  
    {  
        Indoor In[] = new Indoor[3];  
        In[0] = new Indoor("Amit");  
        In[1] = new Indoor("Aniket");  
        In[2] = new Indoor("Archana");  
        System.out.println("Indoor Players...");  
        for(int i=0;i<In.length;i++)  
            In[i].display();  
        System.out.println("Outdoor Players...");  
        Outdoor Out[] = new Outdoor[3];  
        Out[0] = new Outdoor("Anil");  
        Out[1] = new Outdoor("Ameya");  
        Out[2] = new Outdoor("Ajay");  
        for(int i=0;i<Out.length;i++)  
            Out[i].display();  
    }  
}
```

/*

```
C:\>CD C:\MyGames\games  
C:\MyGames\games>javac Indoor.java  
C:\MyGames\games>javac Outdoor.java  
C:\MyGames\games>CD..
```

C:\MyGames>java TestGamesPackage

***/**

Set C:

1. Create a hashtable containing city name & STD code. Display the details of the hashtable. Also search for a specific city and display STD code of that city.

import java.util.Hashtable;

```
public class CityHashtable {  
    public static void main(String[] args) {  
        // Create a hashtable to store city names and STD codes  
        Hashtable<String, String> cityTable = new Hashtable<>();  
  
        // Add some cities and their STD codes to the hashtable  
        cityTable.put("New York", "212");  
        cityTable.put("London", "020");  
        cityTable.put("Paris", "01");  
        cityTable.put("Mumbai", "022");  
  
        // Display the contents of the hashtable  
        System.out.println("City STD codes:");  
        for (String city : cityTable.keySet()) {  
            System.out.println(city + ": " + cityTable.get(city));  
        }  
    }  
}
```

```

// Search for a specific city and display its STD code
String searchCity = "Paris";
if (cityTable.containsKey(searchCity)) {
    System.out.println(searchCity + "'s STD code is " +
cityTable.get(searchCity));
} else {
    System.out.println("City not found");
}
}
}

```

Output :

City STD codes:

New York: 212

London: 020

Paris: 01

Mumbai: 022

Paris's STD code is 01

2. Construct a Linked List containing name: red, blue, yellow and orange.

Then extend your program to do the following: Display the contents of the List using an iterator Display the contents of the List in reverse order using a ListIterator. Create another list containing pink & green. Insert the elements of this list between blue & yellow.

```
import java.util.LinkedList;
```

```
import java.util.ListIterator;
```

```
public class LinkedListExample {
```

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



```
// Create a linked list containing the names "red", "blue", "yellow",  
"orange"
```

```
LinkedList<String> colors = new LinkedList<>();  
colors.add("red");  
colors.add("blue");  
colors.add("yellow");  
colors.add("orange");
```

```
// Display the contents of the linked list using an iterator
```

```
System.out.println("Linked list contents (using iterator:");  
ListIterator<String> iterator = colors.listIterator();  
while (iterator.hasNext()) {  
    System.out.println(iterator.next());  
}
```

```
// Display the contents of the linked list in reverse order using a  
ListIterator
```

```
System.out.println("\nLinked list contents (using ListIterator in reverse  
order:");  
while (iterator.hasPrevious()) {  
    System.out.println(iterator.previous());  
}
```

```
// Create another linked list containing the names "pink" and "green"
```

```
LinkedList<String> moreColors = new LinkedList<>();  
moreColors.add("pink");  
moreColors.add("green");
```

// Insert the elements of the second linked list between "blue" and "yellow" in the first linked list

```
ListIterator<String> iterator1 = colors.listIterator();  
while (iterator1.hasNext()) {  
    if (iterator1.next().equals("blue")) {  
        iterator1.previous();  
        for (String color : moreColors) {  
            iterator1.add(color);  
        }  
        break;  
    }  
}
```

// Display the modified linked list

```
System.out.println("\nModified linked list contents:");  
for (String color : colors) {  
    System.out.println(color);  
}  
}  
}
```

Output :

Linked list contents (using iterator):

red

blue

yellow

orange

Linked list contents (using ListIterator in reverse order):

orange

yellow

blue

red

Modified linked list contents:

red

blue

pink

green

yellow

orange

3. Define an abstract class Staff with members name & address. Define two sub classes FullTimeStaff(Department, Salary) and PartTimeStaff(numberOfHours, ratePerHour). Define appropriate constructors. Create n objects which could be of either FullTimeStaff or PartTimeStaff class by asking the user's choice. Display details of FullTimeStaff and PartTimeStaff.

```
public abstract class Staff {
```

```
    String name;
```

```
    String address;
```

```
    public Staff(String name, String address) {
```

```
        this.name = name;
```

```
        this.address = address;
```

```
}
```

```
public abstract void displayDetails();
```

```
}
```

```
public class FullTimeStaff extends Staff {
```

```
    String department;
```

```
    double salary;
```

```
    public FullTimeStaff(String name, String address, String department, double salary) {
```

```
        super(name, address);
```

```
        this.department = department;
```

```
        this.salary = salary;
```

```
    }
```

```
@Override
```

```
public void displayDetails() {
```

```
    System.out.println("Full-time staff member:");
```

```
    System.out.println("Name: " + name);
```

```
    System.out.println("Address: " + address);
```

```
    System.out.println("Department: " + department);
```

```
    System.out.println("Salary: " + salary);
```

```
}
```

```
}
```

```
public class PartTimeStaff extends Staff {
```

```
int numberOfHours;
```

```
double ratePerHour;
```

```
public PartTimeStaff(String name, String address, int numberOfHours,  
double ratePerHour) {
```

```
    super(name, address);
```

```
    this.numberOfHours = numberOfHours;
```

```
    this.ratePerHour = ratePerHour;
```

```
}
```

```
@Override
```

```
public void displayDetails() {
```

```
    System.out.println("Part-time staff member:");
```

```
    System.out.println("Name: " + name);
```

```
    System.out.println("Address: " + address);
```

```
    System.out.println("Number of hours: " + numberOfHours);
```

```
    System.out.println("Rate per hour: " + ratePerHour);
```

```
}
```

```
}
```

```
Scanner input = new Scanner(System.in);
```

```
System.out.println("Enter 1 for full-time staff or 2 for part-time staff: ");
```

```
int choice = input.nextInt();
```

```
if (choice == 1) {
```

```
    System.out.println("Enter name: ");
```

```
String name = input.next();  
System.out.println("Enter address: ");  
String address = input.next();  
System.out.println("Enter department: ");  
String department = input.next();  
System.out.println("Enter salary: ");  
double salary = input.nextDouble();
```

```
Staff staff = new FullTimeStaff(name, address, department, salary);  
staff.displayDetails();
```

```
} else if (choice == 2) {  
    System.out.println("Enter name: ");  
    String name = input.next();  
    System.out.println("Enter address: ");  
    String address = input.next();  
    System.out.println("Enter number of hours: ");  
    int numberOfHours = input.nextInt();  
    System.out.println("Enter rate per hour: ");  
    double ratePerHour = input.next
```

4. Derive a class Square from class Rectangle. Create one more class Circle. Create an interface with only one method called area(). Implement this interface in all classes. Include appropriate data members and constructors in all classes. Write a java program to accept details of Square, Circle & Rectangle and display the area.

```
public interface Shape {  
    double area();  
}
```

```
public class Rectangle implements Shape {  
    double length;  
    double width;  
  
    public Rectangle(double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
  
    @Override  
    public double area() {  
        return length * width;  
    }  
}
```

```
public class Square extends Rectangle {  
    public Square(double side) {  
        super(side, side);  
    }  
}
```

```
public class Circle implements Shape {  
    double radius;  
  
    public Circle(double radius) {
```

```
    this.radius = radius;  
}
```

```
@Override  
public double area() {  
    return Math.PI * radius * radius;  
}  
}
```

```
Scanner input = new Scanner(System.in);
```

```
System.out.println("Enter 1 for square, 2 for circle, or 3 for rectangle: ");  
int choice = input.nextInt();
```

```
if (choice == 1) {  
    System.out.println("Enter side length: ");  
    double sideLength = input.nextDouble();  
    Shape shape = new Square(sideLength);  
    System.out.println("Area: " + shape.area());  
} else if (choice == 2) {  
    System.out.println("Enter radius: ");  
    double radius = input.nextDouble();  
    Shape shape = new Circle(radius);  
    System.out.println("Area: " + shape.area());  
} else if (choice == 3) {  
    System.out.println("Enter length: ");
```



```
double length = input.nextDouble();  
System.out.println("Enter width: ");  
double width = input.nextDouble();  
Shape shape = new Rectangle(length, width);  
System.out.println("Area: " + shape.area());  
}
```

5. Create a package named Series having three different classes to print series:

i. Fibonacci series

ii. Cube of numbers

iii. Square of numbers

Write a java program to generate 'n' terms of the above series.
package Series;

```
public class FibonacciSeries {  
    public static void print(int n) {  
        int a = 0, b = 1;  
        System.out.print("Fibonacci series: ");  
        for (int i = 1; i <= n; i++) {  
            System.out.print(a + " ");  
            int c = a + b;  
            a = b;  
            b = c;  
        }  
        System.out.println();  
    }  
}
```

```
package Series;
```

```
public class CubeSeries {  
    public static void print(int n) {  
        System.out.print("Cube series: ");  
        for (int i = 1; i <= n; i++) {  
            System.out.print(i*i*i + " ");  
        }  
        System.out.println();  
    }  
}
```

```
package Series;
```

```
public class SquareSeries {  
    public static void print(int n) {  
        System.out.print("Square series: ");  
        for (int i = 1; i <= n; i++) {  
            System.out.print(i*i + " ");  
        }  
        System.out.println();  
    }  
}
```

```
import Series.*;
```

```
Scanner input = new Scanner(System.in);
```

```
System.out.println("Enter number of terms: ");
```

```
int n = input.nextInt();
```

```
FibonacciSeries.print(n);
```

```
CubeSeries.print(n);
```

```
SquareSeries.print(n);
```

Assignment No. 4 : File and Exception Handling

Set A:

1. Write a java program to count the number of integers from a given list.(Use command line arguments).

```
public class Main {  
    public static void main(String[] args) {  
        int count = 0;  
        for (String arg : args) {  
            try {  
                // Try to parse the string as an integer  
                int num = Integer.parseInt(arg);  
                count++;  
            } catch (NumberFormatException e) {  
                // Ignore non-integer strings  
            }  
        }  
        System.out.println("Number of integers: " + count);  
    }  
}  
  
java Main 1 2 3 4 5
```

Number of integers: 5

2. Write a java program to check whether given candidate is eligible for voting or not. Handle user defined as well as system defined Exception.

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

```
class Voting
```

```
{
```

```
    public static void main(String args[])
```

```
    {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter your Name: ");
```

```
        String name=sc.nextLine();
```

```
        System.out.println("Enter your age: ");
```

```
        int age=sc.nextInt();
```

```
        if((age>=18)&&(age<=100))
```

```
        {
```

```
            System.out.println("Congratulation "+name+", You are  
eligible for Voting");
```

```
        }
```

```
        else
```

```
        {
```

```
            System.out.println("Sorry "+name+", You are not eligible for  
voting");
```

```
        }
```

```
    }
```

```
}
```

OUTPUT:

```
CA Command Prompt
C:\Java>javac Voting.java
C:\Java>java Voting
Enter your Name:
Vijay
Enter your age:
36
Congratulation Vijay, You are eligible for Voting
C:\Java>
```

```
CA Command Prompt
C:\Java>javac Voting.java
C:\Java>java Voting
Enter your Name:
Aayush
Enter your age:
15
Sorry Aayush, You are not eligible for voting
```

3. Write a java program to calculate the size of a file.

```
import java.io.File;
```

```
public class Main {
```

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

```
        // Get the file path from the command line argument
```

```
        String filePath = args[0];
```

```
        File file = new File(filePath);
```

```
        long size = file.length();
```

```
        System.out.println("Size of file: " + size + " bytes");
```

```
    }
```

```
}
```

OUTPUT :

java Main /path/to/file.txt

4. Write a java program to accept a number from a user, if it is zero then throw user defined Exception "Number is Zero". If it is non-numeric then generate an error "Number is Invalid" otherwise check whether it is palindrome or not.

import java.io.*;

class Numberiszero extends Exception{}

class Slip30A{

public static void main(String args[]){

int r,sum=0,temp;

int n;

DataInputStream dr = new DataInputStream(System.in);

try {

System.out.print("Enter Number : ");

n = Integer.parseInt(dr.readLine());

```
if(n==0){

    throw new Numberiszero();

}else{

    temp=n;

    while(n>0){

        r=n%10;

        sum=(sum*10)+r;

        n=n/10;

    }

    if(temp==sum){

        System.out.println("Palindrome Number ");

    }else{
```



```
        System.out.println("Not Palindrome");
```

```
    }
```

```
}
```

```
} catch (Numberiszero nz) {
```

```
    System.out.println("Number is Zero");
```

```
}
```

```
catch (NumberFormatException e){
```

```
    System.out.println("Number is Invalid");
```

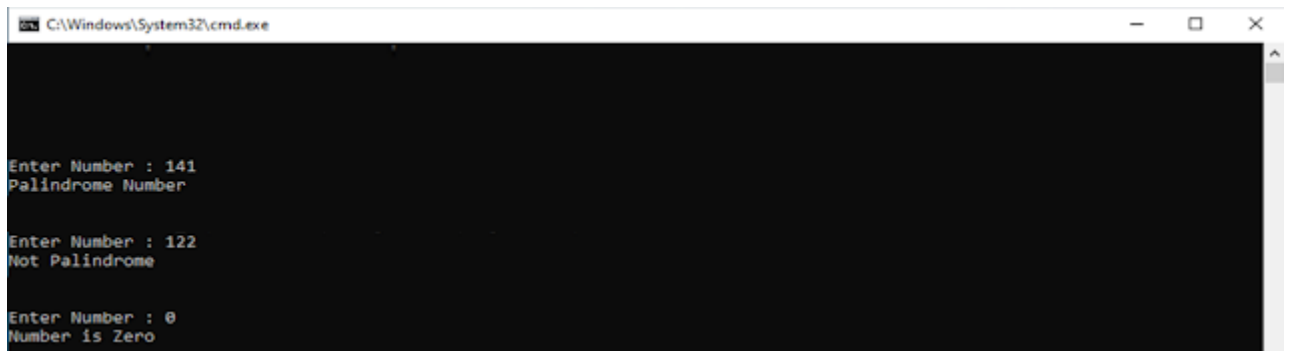
```
}
```

```
catch (Exception e){}
```

```
}
```

```
}
```

Output :



```
C:\Windows\System32\cmd.exe

Enter Number : 141
Palindrome Number

Enter Number : 122
Not Palindrome

Enter Number : 0
Number is Zero
```

5. Write a java program to accept a number from user, If it is greater than 100 then throw user defined exception “Number is out of Range” otherwise do the addition of digits of that number. (Use static keyword)

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

```
public class Main {
```

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

```
        Scanner scan = new Scanner(System.in);
```

```
        System.out.print("Enter a number: ");
```

```
        int num = scan.nextInt();
```

```
        // Check if the number is greater than 100
```

```
        if (num > 100) {
```

```
            throw new NumberOutOfRangeException("Number is out of range");
```

```
        }
```

```
// Perform the addition of the digits of the number
int sum = 0;
while (num > 0) {
    sum += num % 10;
    num /= 10;
}
System.out.println("Sum of digits: " + sum);
}
}
```

```
class NumberOutOfRangeException extends Exception {
    public NumberOutOfRangeException(String message) {
        super(message);
    }
}
```

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

```
public class Main {
    private static Scanner scan = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.print("Enter a number: ");
        int num = scan.nextInt();
    }
}
```

```
// Check if the number is greater than 100
if (num > 100) {
    throw new NumberOutOfRangeException("Number is out of
range");
}
```

```
// Perform the addition of the digits of the number
int sum = 0;
while (num > 0) {
    sum += num % 10;
    num /= 10;
}
System.out.println("Sum of digits: " + sum);
}
}
```

```
class NumberOutOfRangeException extends Exception {
    public NumberOutOfRangeException(String message) {
        super(message);
    }
}
```

Set B:

1. Write a java program to copy the data from one file into another file, while copying change the case of characters in target file and replaces all digits by '*' symbol.

/*To copy the content one file into the another file while copying change the case of alphabet and replase all Digit by '*'.

*/

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

```
class Slip2
```

```
{
```

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

```
    {
```

```
        int c;
```

```
        try
```

```
        {
```

```
            FileReader fr=new FileReader("a.txt");
```

```
            FileWriter fw=new FileWriter("b.txt");
```

```
            while((c=fr.read())!=-1)
```

```
            {
```

```
                if(c>=65&& c<=90)
```

```
                {
```

```
                    c=c+32;
```

```
                    fw.write(c);
```

```
                }
```

```
                else if(c>=97&& c<=122)
```

```
                {
```

```
                    c=c-32;
```

```
                    fw.write(c);
```

```
                }
```

```
                else if(c>=48&& c<=57)
```

```
                {
```

```

        fw.write('*');
    }
    else
    {
        fw.write(c);
    }

}
System.out.println("Copy Successfully");
fr.close();
fw.close();
}catch(Exception e)
{
    System.out.println(e);
}
}
}

```

OUTPUT:

/*Output:-

a.txtfile

Shinde Sachin 9767@gmail.com

1234R

D:\javapro\file>javac Slip2.java

D:\javapro\file>java Slip2

Copy Successfully

D:\javapro\file>

b.txt

sSHINDE sACHIN **@GMAIL.COM**

******r**

***/**

2. Write a java program to accept string from a user. Write ASCII values of the characters from a string into the file.

```
public class AsciiValue {  
  
    public static void main(String[] args) {  
  
        char ch = 'a';  
        int ascii = ch;  
        // You can also cast char to int  
        int castAscii = (int) ch;  
  
        System.out.println("The ASCII value of " + ch + " is: " + ascii);  
        System.out.println("The ASCII value of " + ch + " is: " +  
castAscii);  
    }  
}
```

OUTPUT:

The ASCII value of a is: 97

The ASCII value of a is: 97

3. Write a java program to accept a number from a user, if it less than 5 then throw user defined Exception “Number is small”, if it is greater than 10 then throw user defined exception “Number is Greater”, otherwise calculate its factorial.

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

```
public class Main {
```

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

```
        Scanner scan = new Scanner(System.in);
```

```
        System.out.print("Enter a number: ");
```

```
        int num = scan.nextInt();
```

```
        // Check if the number is less than 5
```

```
        if (num < 5) {
```

```
            throw new NumberTooSmallException("Number is small");
```

```
        }
```

```
        // Check if the number is greater than 10
```

```
        if (num > 10) {
```

```
            throw new NumberTooLargeException("Number is greater");
```

```
        }
```

```
        // Calculate the factorial of the number
```

```
        int factorial = 1;
```

```
        for (int i = 1; i <= num; i++) {
```



```
        factorial *= i;
    }
    System.out.println("Factorial: " + factorial);
}
}
```

```
class NumberTooSmallException extends Exception {
    public NumberTooSmallException(String message) {
        super(message);
    }
}
```

```
class NumberTooLargeException extends Exception {
    public NumberTooLargeException(String message) {
        super(message);
    }
}
```

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

```
public class Main {
    private static Scanner scan = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.print("Enter a number: ");
    }
}
```

```
int num = scan.nextInt();

// Check if the number is less than 5
if (num < 5) {
    throw new NumberTooSmallException("Number is small");
}

// Check if the number is greater than 10
if (num > 10) {
    throw new NumberTooLargeException("Number is greater");
}

// Calculate the factorial of the number
int factorial = 1;
for (int i = 1; i <= num; i++) {
    factorial *= i;
}

System.out.println("Factorial: " + factorial);
}
```

```
class NumberTooSmallException extends Exception {
    public NumberTooSmallException(String message) {
        super(message);
    }
}
```

```
}  
}
```

```
class NumberTooLargeException extends Exception {  
    public NumberTooLargeException(String message) {  
        super(message);  
    }  
}
```

4. Write a java program to display contents of a file in reverse order.

```
import java.io.*;  
import java.util.*;  
class FileReverse  
{  
    File f,f2;  
    String str;  
    FileReverse()  
    {  
        Scanner s=new Scanner(System.in);  
        System.out.println("Enter the source path: ");  
        String path=s.nextLine();  
        System.out.println("Enter the source name: ");  
        String name=s.nextLine();  
        f=new File(path,name);
```

```

    }
    void reverse()
    {
        try
        {
            RandomAccessFile file = new
RandomAccessFile(f,"r");
            long n= f.length()-1;
            int i=0;
            while (n>=0)
            {
                if (n== -1)
                    break;
                else
                {
                    file.seek(n);
                    i= file.read();
                    n=n-1;
                    System.out.print((char)i);
                }
            }
            file.close();
        }catch(Exception e)
        {

```

```

        System.out.print(e);
    }
}

class TestFileReverse
{
    public static void main(String args[])
    {
        FileReverse obj=new FileReverse();
        obj.reverse();
    }
}

```

5. Write a java program to display each word from a file in reverse order.

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class ReverseWords {
    public static void main(String[] args) throws IOException {
        // Prompt the user to enter the file name
        System.out.print("Enter a file name: ");
        Scanner input = new Scanner(System.in);
    }
}

```

```
String fileName = input.nextLine();

// Open the file
FileReader fileReader = new FileReader(fileName);
BufferedReader reader = new BufferedReader(fileReader);

// Read each line of the file
String line;
while ((line = reader.readLine()) != null) {
    // Split the line into words
    String[] words = line.split("\\s+");

    // Print each word in reverse order
    for (int i = words.length - 1; i >= 0; i--) {
        System.out.print(words[i] + " ");
    }
    System.out.println();
}

// Close the file
reader.close();
}
}
```

Set C:

1. Write a java program to accept list of file names through command line. Delete the files having extension .txt. Display name, location and size of remaining files.

```
import java.io.*;
class Slip12
{
    public static void main(String args[]) throws Exception
    {
        for(int i=0;i {
            File file=new File(args[i]);
            if(file.isFile())
            {
                String name = file.getName();
                if(name.endsWith(".txt"))
                {
                    file.delete();
                    System.out.println("file is deleted " +file);
                }
            }
            else
            {
                System.out.println(name + " "+file.length()+" bytes")
            }
        }
        else
        {
            System.out.println(args[i]+ "is not a file");
        }
    }
}
```

2. Write a java program to display the files having extension .txt from a given directory.

```
import java.io.File;

class Main {

    public static void main(String[] args) {
        File directory = new File("Directory");

        // list all files present in the directory
        File[] files = directory.listFiles();
        System.out.println("Files\t\t\tExtension");

        for(File file : files) {
            // convert the file name into string
            String fileName = file.toString();

            int index = fileName.lastIndexOf('.');
            if(index > 0) {
                String extension = fileName.substring(index + 1);
                System.out.println(fileName + "\t" + extension);
            }
        }
    }
}
```

Output:

```
import java.io.File;

class Main {

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



```

File directory = new File("Directory");

// list all files present in the directory
File[] files = directory.listFiles();
System.out.println("Files\t\t\tExtension");

for(File file : files) {
    // convert the file name into string
    String fileName = file.toString();

    int index = fileName.lastIndexOf('.');
    if(index > 0) {
        String extension = fileName.substring(index + 1);
        System.out.println(fileName + "\t" + extension);
    }
}

}

```

3. Write a java program to count number of lines, words and characters from a given file.

```

// Java program to count the
// number of lines, words, sentences,
// characters, and whitespaces in a file
import java.io.*;

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

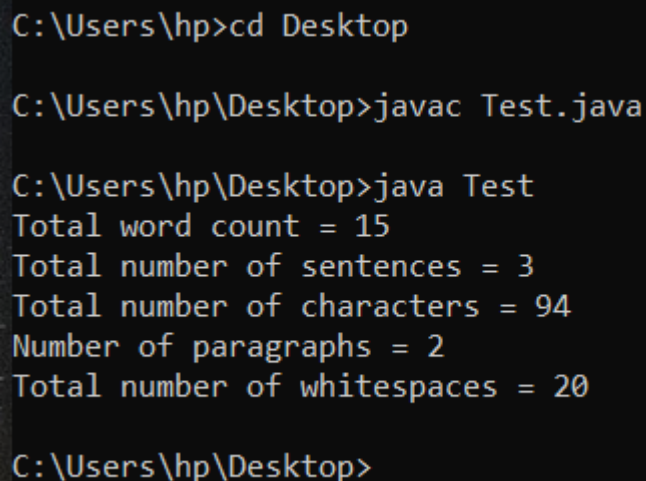
```

```
File file = new
File("C:\\Users\\hp\\Desktop\\TextReader.txt");
FileInputStream fileInputStream = new
FileInputStream(file);
InputStreamReader inputStreamReader = new
InputStreamReader(fileInputStream);
BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);
```

```
String line;
int wordCount = 0;
int characterCount = 0;
int paraCount = 0;
int whiteSpaceCount = 0;
int sentenceCount = 0;
```

```
while ((line = bufferedReader.readLine()) != null) {
    if (line.equals("")) {
        paraCount += 1;
    }
    else {
        characterCount += line.length();
        String words[] = line.split("\\s+");
        wordCount += words.length;
        whiteSpaceCount += wordCount - 1;
        String sentence[] = line.split("[!?.:]+");
        sentenceCount += sentence.length;
    }
}
if (sentenceCount >= 1) {
    paraCount++;
}
```

```
        System.out.println("Total word count = "+
wordCount);
        System.out.println("Total number of sentences =
"+ sentenceCount);
        System.out.println("Total number of characters =
"+ characterCount);
        System.out.println("Number of paragraphs = "+
paraCount);
        System.out.println("Total number of whitespaces =
"+ whiteSpaceCount);
    }
}
```



```
C:\Users\hp>cd Desktop
C:\Users\hp\Desktop>javac Test.java
C:\Users\hp\Desktop>java Test
Total word count = 15
Total number of sentences = 3
Total number of characters = 94
Number of paragraphs = 2
Total number of whitespaces = 20
C:\Users\hp\Desktop>
```

4. Write a java program to read the characters from a file, if a character is alphabet then reverse its case, if not then display its category on the Screen. (whether it is Digit or Space)

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

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

```
class MyFile
```

```
{
```

```
    File f1;
```

```
    MyFile()
```

```
    {
```

```
        Scanner s = new Scanner(System.in);
```

```
        System.out.print("Enter the path of file: ");
```

```
        String path= s.nextLine();
```

```
        System.out.print("Enter the name of file: ");
```

```
        String name= s.nextLine();
```

```
        f1=new File(path, name);
```

```
    }
```

```
    void check()
```

```
    {
```

```
        try
```

```
        {
```

```
            if(f1.isFile() && f1.exists())
```

```
            {
```

```
                FileInputStream fobj = new
```

```
FileInputStream(f1);
```

```

int i=fobj.read();
while(i!=-1)
{
    if(Character.isDigit((char)i))
    {
        System.out.println("Character is
digit");
    }
    else if (Character.isSpace((char)i))
    {
        System.out.println("Character is
Space");
    }
    else
    {
        if(Character.isUpperCase((char)i))

System.out.println(Character.toLowerCase((char)i));
        else

System.out.println(Character.toUpperCase((char)i));
    }
    i=fobj.read();
}
}

```

```

        else
        {
            System.out.println("File doesn't exist");
        }
    }
    catch(Exception e)
    {
        System.out.print(e);
    }
}
}
class slip1
{
    public static void main(String args[])
    {
        MyFile obj = new MyFile();
        obj.check();
    }
}

```

5. Write a java program to validate PAN number and Mobile Number. If it is invalid then throw user defined Exception “Invalid Data”, otherwise display it.

```

import java.util.regex.Matcher;
import java.util.regex.Pattern;

```

```
public class ValidateData {  
    // Regular expression for a valid PAN number (alphanumeric, 10  
    characters)  
    private static final String PAN_REGEX = "[A-Za-z0-9]{10}$";  
  
    // Regular expression for a valid mobile number (numeric, 10  
    digits)  
    private static final String MOBILE_REGEX = "[0-9]{10}$";  
  
    // User-defined exception class  
    static class InvalidDataException extends Exception {  
        public InvalidDataException(String message) {  
            super(message);  
        }  
    }  
  
    public static void main(String[] args) {  
        // Test data  
        String[] panNumbers = { "ABCDE1234Z", "1234567890",  
"ABCDE12345" };  
  
        String[] mobileNumbers = { "1234567890", "12345678901",  
"12345678A" };  
  
        for (String pan : panNumbers) {
```

```

    try {
        validatePAN(pan);

        System.out.println("Valid PAN: " + pan);
    } catch (InvalidDataException e) {

        System.out.println("Invalid PAN: " + pan + " (" +
e.getMessage() + ")");
    }
}

for (String mobile : mobileNumbers) {
    try {
        validateMobile(mobile);

        System.out.println("Valid mobile: " + mobile);
    } catch (InvalidDataException e) {

        System.out.println("Invalid mobile: " + mobile + " (" +
e.getMessage() + ")");
    }
}
}

```

```

// Validates a PAN number and throws an exception if it is invalid
public static void validatePAN(String pan) throws
InvalidDataException {
    if (!isValid(pan, PAN_REGEX)) {
        throw new InvalidDataException("Invalid PAN format");
    }
}

```



```
}  
}
```

// Validates a mobile number and throws an exception if it is invalid

```
public static void validateMobile(String mobile) throws  
InvalidDataException {  
    if (!isValid(mobile, MOBILE_REGEX)) {  
        throw new InvalidDataException("Invalid mobile format");  
    }  
}
```

// Returns true if the given string matches the given regular expression

```
public static boolean isValid(String s, String regex) {  
    Pattern pattern = Pattern.compile(regex);  
    Matcher matcher = pattern.matcher(s);  
    return matcher.matches();  
}  
}
```

Assignment No. 5: Applet, AWT, Event and Swing Programming

Set A:

1. Write a program that asks the user's name, and then greets the user by name. Before outputting the user's name, convert it to upper case letters. For example, if the user's name is Raj, then the program should respond "Hello, RAJ, nice to meet you!".

```
import java.util.Scanner; // Import the Scanner class
```

```
public class Greeting {
```

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

```
        Scanner reader = new Scanner(System.in); // Create a Scanner  
        object to read input from the user
```

```
        System.out.print("Enter your name: "); // Prompt the user to  
        enter their name
```

```
        String name = reader.nextLine(); // Read the user's name
```

```
        name = name.toUpperCase(); // Convert the name to upper  
        case letters
```

```
        System.out.println("Hello, " + name + ", nice to meet you!"); //  
        Greet the user by name
```

```
    }
```

```
}
```

To run this program, you will need to have the Java Development Kit (JDK) installed on your computer. You can then use a tool such as the Java compiler (javac) to compile the program, and the Java interpreter (java) to run the compiled program.

For example, to compile and run the program, you can use the following commands:

```
javac Greeting.java
```

```
java Greeting
```

The program will then prompt the user to enter their name, and will greet the user by name after the user enters their name.

2. Write a program that reads one line of input text and breaks it up into words. The words should be output one per line. A word is defined to be a sequence of letters. Any characters in the input that are not letters should be discarded. For example, if the user inputs the line He said, "That's not a good idea." then the output of the program should be

He

said

thats

not

a

good

idea

```
import textio.TextIO;
```

```
public class ListWordsInString {
```

```
public static void main(String[] args) {  
  
    String line;  
    int i;  
    char ch;  
    boolean didCR;  
  
    System.out.println("Enter a line of text.");  
    System.out.print("? ");  
    line = TextIO.getln();  
  
    System.out.println();  
    didCR = true;  
  
    for ( i = 0; i < line.length(); i++ ) {  
        ch = line.charAt(i);  
        if ( Character.isLetter(ch) ) {  
            System.out.print(ch);  
            didCR = false;  
        }  
        else {  
            if ( didCR == false ) {  
                System.out.println();  
                didCR = true;  
            }  
        }  
    }  
}
```

```
    }  
  }  
}
```

```
System.out.println();
```

```
}
```

```
}
```

Output:

He

said

That

s

not

a

good

idea

3. Write a program that will read a sequence of positive real numbers entered by the user and will print the same numbers in sorted order from smallest to largest. The user will input a zero to mark the end of the input. Assume that at most 100 positive numbers will be entered.

```
import java.util.Scanner; // Import the Scanner class
```

```
public class SortNumbers {  
    public static void main(String[] args) {  
        Scanner reader = new Scanner(System.in); // Create a Scanner  
        object to read input from the user  
  
        // Declare an array to store the numbers  
        double[] numbers = new double[100];  
  
        // Read the numbers from the user  
        int count = 0;  
        System.out.print("Enter a number (0 to end): ");  
        double number = reader.nextDouble();  
        while (number != 0) {  
            numbers[count] = number;  
            count++;  
            System.out.print("Enter a number (0 to end): ");  
            number = reader.nextDouble();  
        }  
  
        // Sort the numbers  
        for (int i = 0; i < count - 1; i++) {  
            for (int j = i + 1; j < count; j++) {  
                if (numbers[i] > numbers[j]) {  
                    double temp = numbers[i];
```

```

        numbers[i] = numbers[j];
        numbers[j] = temp;
    }
}
}

// Print the sorted numbers
System.out.println("Sorted numbers:");
for (int i = 0; i < count; i++) {
    System.out.println(numbers[i]);
}
}
}

```

To run this program, you will need to have the Java Development Kit (JDK) installed on your computer. You can then use a tool such as the Java compiler (javac) to compile the program, and the Java interpreter (java) to run the compiled program.

For example, to compile and run the program, you can use the following commands:

```
javac SortNumbers.java
```

```
java SortNumbers
```

The program will then prompt the user to enter a number, and will continue reading numbers until the user enters a zero. The program will then sort the numbers and print them in sorted order.

4. Create an Applet that displays the x and y position of the cursor movement using Mouse and Keyboard. (Use appropriate listener).

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
public class CursorTracker extends Applet implements  
MouseListener, KeyListener {
```

```
// Declare variables to store the current x and y position of the  
cursor
```

```
int xPos = 0;
```

```
int yPos = 0;
```

```
// Declare a label to display the current x and y position
```

```
Label positionLabel;
```

```
public void init() {
```

```
// Set the background color to white
```

```
setBackground(Color.white);
```

```
// Add the mouse and keyboard listeners
```

```
addMouseListener(this);
```

```
addKeyListener(this);
```

```
// Create a label to display the current x and y position
```



```
positionLabel = new Label("X: 0, Y: 0");

// Add the label to the applet
add(positionLabel);
}

// MouseListener methods
public void mousePressed(MouseEvent e) {
    // Update the x and y position when the mouse is pressed
    xPos = e.getX();
    yPos = e.getY();

    // Update the label to display the new x and y position
    positionLabel.setText("X: " + xPos + ", Y: " + yPos);
}

public void mouseReleased(MouseEvent e) {}
public void mouseClicked(MouseEvent e) {}
public void mouseEntered(MouseEvent e) {}
public void mouseExited(MouseEvent e) {}

// KeyListener methods
public void keyPressed(KeyEvent e) {
    // Update the x and y position when a key is pressed
```

```

xPos = e.getX();
yPos = e.getY();

// Update the label to display the new x and y position
positionLabel.setText("X: " + xPos + ", Y: " + yPos);
}

public void keyReleased(KeyEvent e) {}
public void keyTyped(KeyEvent e) {}
}

```

To use this applet, you will need to include it in an HTML file and run it in a Java-enabled browser. For example:

```

<html>

<body>

    <applet code="CursorTracker.class" width="300" height="300">
    </applet>

</body>
</html>

```

This applet will display the current x and y position of the cursor in a label, and update the position whenever the mouse is clicked or a key is pressed.

5. Create the following GUI screen using appropriate layout managers.

```
import java.awt.*;
```

```
import java.awt.event.*;
import javax.swing.*;

public class NumberAdder extends JFrame {
    // Declare text fields for the two numbers and the result
    JTextField number1Field, number2Field, resultField;

    public NumberAdder() {
        // Set the title and layout of the frame
        setTitle("Number Adder");
        setLayout(new BorderLayout());

        // Create a panel for the first number
        JPanel panel1 = new JPanel();
        panel1.add(new JLabel("First number:"));
        number1Field = new JTextField(10);
        panel1.add(number1Field);

        // Create a panel for the second number
        JPanel panel2 = new JPanel();
        panel2.add(new JLabel("Second number:"));
        number2Field = new JTextField(10);
        panel2.add(number2Field);
```

```
// Create a panel for the result
JPanel panel3 = new JPanel();
panel3.add(new JLabel("Result:"));
resultField = new JTextField(10);
resultField.setEditable(false);
panel3.add(resultField);

// Add the panels to the frame
add(panel1, BorderLayout.NORTH);
add(panel2, BorderLayout.CENTER);
add(panel3, BorderLayout.SOUTH);

// Create a button to perform the addition
JButton addButton = new JButton("Add");
add(addButton, BorderLayout.EAST);

// Add an action listener to the button
addButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Get the values from the text fields
        int number1 = Integer.parseInt(number1Field.getText());
        int number2 = Integer.parseInt(number2Field.getText());

        // Calculate the result
```

```
int result = number1 + number2;

// Display the result in the result field
resultField.setText(Integer.toString(result));
}
});
}

public static void main(String[] args) {
    NumberAdder frame = new NumberAdder();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
}
}

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class NumberAdder extends JFrame {
    // Declare text
```

Set B:

- 1. Write a java program to implement a simple arithmetic calculator. Perform appropriate validations.**

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class Calc implements ActionListener
{
    char o;
    int ctr=0;
    String value="";
    Double answer, v1, v2;
    JFrame f;
    private JTextField tField;
    private JButton num0, num1, num2, num3, num4, num5,
num6, num7, num8,num9;
    private JButton bAdd, bSub, bMul, bDiv, bDot,equals;
    Calc()
    {
        f = new JFrame("Simple Calculator");
        tField = new JTextField(15);
        num0 = new JButton ("0");
        num1 = new JButton ("1");
        num2 = new JButton ("2");
        num3 = new JButton ("3");
        num4 = new JButton ("4");
        num5 = new JButton ("5");
        num6 = new JButton ("6");
        num7 = new JButton ("7");
        num8 = new JButton ("8");
        num9 = new JButton ("9");
        bAdd = new JButton ("+");
```

```
bSub = new JButton ("-");
bMul = new JButton ("*");
bDiv = new JButton ("/");
bDot = new JButton (".");
equals = new JButton("=");
tField.setText("0.");
tField.setEnabled(false);
    f.add(tField);
    f.add(num1);
    f.add(num2);
    f.add(num3);
    f.add(bAdd);
    f.add(num4);
    f.add(num5);
f.add(num6);
f.add(bSub);
f.add(num7);
    f.add(num8);
    f.add(num9);
    f.add(bMul);
    f.add(num0);
    f.add(bDot);
    f.add(equals);
    f.add(bDiv);
f.setLayout(new FlowLayout());
f.setDefaultCloseOperation(f.EXIT_ON_CLOSE);
    f.setResizable(false);
f.setSize(200,220);
f.setVisible(true);
num0.addActionListener(this);
num1.addActionListener(this);
num2.addActionListener(this);
num3.addActionListener(this);
```

```
num4.addActionListener(this);
num5.addActionListener(this);
num6.addActionListener(this);
num7.addActionListener(this);
num8.addActionListener(this);
num9.addActionListener(this);
bAdd.addActionListener(this);
bSub.addActionListener(this);
bMul.addActionListener(this);
bDiv.addActionListener(this);
bDot.addActionListener(this);
equals.addActionListener(this);
}
public void actionPerformed(ActionEvent a)
{
    try
    {
        if(a.getSource()==num0)
        {
            value+=0;
            tField.setText(value);
        }
        if(a.getSource()==num1)
        {
            value+=1;
            tField.setText(value);
        }
        if(a.getSource()==num2)
        {
            value+=2;
            tField.setText(value);
        }
        if(a.getSource()==num3)
```



```
{
    value+=3;
    tField.setText(value);
}
if(a.getSource()==num4)
{
    value+=4;
    tField.setText(value);
}
if(a.getSource()==num5)
{
    value+=5;
    tField.setText(value);
}
if(a.getSource()==num6)
{
    value+=6;
    tField.setText(value);
}
if(a.getSource()==num7)
{
    value+=7;
    tField.setText(value);
}
if(a.getSource()==num8)
{
    value+=8;
    tField.setText(value);
}
if(a.getSource()==num9)
{
    value+=9;
    tField.setText(value);
}
```

```
}  
if (a.getSource() == bAdd)  
{  
    v1 = Double.parseDouble( tField.getText() );  
    ctr=0;  
    o = '+';  
    value="";  
    tField.setText(o+value);  
}  
if (a.getSource() == bSub)  
{  
    v1 = Double.parseDouble( tField.getText() );  
    ctr=0;  
    o = '-';  
    value="";  
    tField.setText(o+value);  
}  
if (a.getSource() == bMul)  
{  
  
    v1 = Double.parseDouble( tField.getText() );  
    ctr=0;  
    o = '*';  
    value="";  
    tField.setText(o+value);  
}  
if (a.getSource() == bDiv)  
{  
  
    v1 = Double.parseDouble( tField.getText() );  
    ctr=0;  
    o = '/';  
    value="";
```

```

        tField.setText(o+value);
    }
    if(a.getSource()==equals)
    {
        value="";
        v2 = Double.parseDouble(tField.getText());
        if(o=='+')
        {
            ctr=0;
            answer = v1 + v2;
            tField.setText("" +answer);
            value=""; v1=null; v2=null;
        }
        else if(o=='-')
        {
            ctr=0;

            answer = v1 - v2;
            tField.setText("" +answer);
            value=""; v1=null; v2=null;
        }
        else if(o=='*')
        {
            ctr=0;
            answer = v1 * v2;
            tField.setText("" +answer);
            value=""; v1=null; v2=null;
        }
        else if(o=='/')
        {
            ctr=0;
            answer = v1 / v2;
            tField.setText("" +answer);

```

```

        value=""; v1=null; v2=null;
    }
}
if(a.getSource() == bDot)
{
    if(ctr==0)
    {
        value+=".";
        ctr+=1;
        tField.setText("" +value);
    }
    else
    {
        System.out.print("");
    }
}
}
catch(NumberFormatException nfe)
{
    JOptionPane.showMessageDialog(null,"Number Format
Exception","Exception",JOptionPane.WARNING_MESSAGE);
}
catch(StringIndexOutOfBoundsException str)
{
    JOptionPane.showMessageDialog(null,"String Index Out
of Bounds
Exception","Exception",JOptionPane.WARNING_MESSAGE);
}
catch(NullPointerException str)
{
    JOptionPane.showMessageDialog(null,"Null Pointer

```

```

Exception", "Exception", JOptionPane.WARNING_MESSAGE);
    }
    } // END OF ACTION EVENTS
    public static void main (String args[])
    {
        Calc s = new Calc();
    }
}

```

2. Write a java program to implement following. Program should handle appropriate events.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.undo.*;

```

```

public class TextEditor extends JFrame {
    private JTextArea textArea;
    private UndoManager undoManager;

```

```

    public TextEditor() {
        super("Text Editor");

```

```

        textArea = new JTextArea();
        textArea.setLineWrap(true);
        textArea.setWrapStyleWord(true);

```

```

        // Create the undo manager
        undoManager = new UndoManager();
        textArea.getDocument().addUndoableEditListener(new
UndoableEditListener() {
            public void undoableEditHappened(UndoableEditEvent e) {

```

```
        undoManager.addEdit(e.getEdit());
    }
});

// Create the toolbar
JToolBar toolBar = new JToolBar();
toolBar.setFloatable(false);

// Add the undo button
JButton undoButton = new JButton("Undo");
undoButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            undoManager.undo();
        } catch (CannotUndoException ex) {
            ex.printStackTrace();
        }
    }
});
toolBar.add(undoButton);

// Add the redo button
JButton redoButton = new JButton("Redo");
redoButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            undoManager.redo();
        } catch (CannotRedoException ex) {
            ex.printStackTrace();
        }
    }
});
toolBar.add(redoButton);
```

```
// Add the cut button
JButton cutButton = new JButton("Cut");
cutButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textArea.cut();
    }
});
toolBar.add(cutButton);

// Add the copy button
JButton copyButton = new JButton("Copy");
copyButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textArea.copy();
    }
});
toolBar.add(copyButton);

// Add the paste button
JButton pasteButton = new JButton("Paste");
pasteButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textArea.paste();
    }
});
toolBar.add(pasteButton);

// Add the text area and toolbar to the frame
add(toolBar, BorderLayout.NORTH);
add(new JScrollPane(textArea), BorderLayout.CENTER);

setSize(500, 300);
```

```
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setLocationRelativeTo(null);  
}
```

```
public static void main(String[]
```

3. Write an applet application to draw Temple.

```
import java.applet.Applet;  
import java.awt.*;  
    public class Slip10 extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawRect(100,150,90,120);  
        g.drawRect(130,230,20,40);  
        g.drawLine(150,100,100,150);  
        g.drawLine(150,100,190,150);  
        g.drawLine(150,50,150,100);  
        g.drawRect(150,50,20,20);  
    }  
}
```


4. Write an applet application to display Table lamp. The color of lamp should get change in random color.

```
import java.awt.*;
import java.applet.*;

public class Lamp extends Applet
{
    public void paint(Graphics g)
    {
        g.setColor(Color.BLUE);
        g.fillRoundRect(90,250,120,85,30,30);           // Lamp
        Platform

        g.setColor(Color.MAGENTA);
        g.drawLine(125,250,125,100);
        g.drawLine(175,250,175,100) ;                 // the base of the
        lamp

        g.setColor(Color.ORANGE);
        g.fillOval(85,161,135,40);
        g.drawArc(85,87,130,50,62,58);                // the lamp shade, top
        and bottom edges

        g.setColor(Color.PINK);
        g.drawLine(85,177,119,89);
        g.drawLine(215,177,181,89);                    // lamp shade, sides

        g.setColor(Color.BLUE);
        g.fillArc(78,120,40,40,63,-174);
        g.setColor(Color.RED);                          // dots on the shade
        g.fillOval(120,96,40,40);
        g.setColor(Color.GREEN);
```

```
g.fillArc(173,100,40,40,110,180);  
}  
  
}
```

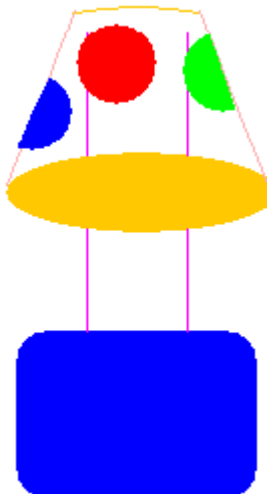
Compile this java file using command `javac Lamp.java`

Now Create the HTML file as shown below

```
<html>  
<head><title>Creating Lamp </title> </head>  
<body>  
<applet code="Lamp.class" width=400 height=400>  
</applet>  
</body>
```

Execute this program using the command `appletviewer Lamp.html`

You Will get the Output as shown below :



5. Write a java program to design email registration form.(Use maximum Swing component in form).

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Registration extends JFrame implements
ActionListener
{
    JLabel l1, l2, l3, l4, l5, l6, l7, l8;
    JTextField tf1, tf2, tf5, tf6, tf7;
    JButton btn1, btn2;
    JPasswordField p1, p2;
    Registration()
    {
        setVisible(true);
        setSize(700, 700);
        setLayout(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Registration Form in Java");
        l1 = new JLabel("Registration Form in Windows Form:");
        l1.setForeground(Color.blue);
        l1.setFont(new Font("Serif", Font.BOLD, 20));
        l2 = new JLabel("Name:");
        l3 = new JLabel("Email-ID:");
        l4 = new JLabel("Create Passowrd:");
        l5 = new JLabel("Confirm Password:");
        l6 = new JLabel("Country:");
        l7 = new JLabel("State:");
        l8 = new JLabel("Phone No:");
        tf1 = new JTextField();
        tf2 = new JTextField();
```

```
p1 = new JPasswordField();
p2 = new JPasswordField();
tf5 = new JTextField();
tf6 = new JTextField();
tf7 = new JTextField();
btn1 = new JButton("Submit");
btn2 = new JButton("Clear");
btn1.addActionListener(this);
btn2.addActionListener(this);
l1.setBounds(100, 30, 400, 30);
l2.setBounds(80, 70, 200, 30);
l3.setBounds(80, 110, 200, 30);
l4.setBounds(80, 150, 200, 30);
l5.setBounds(80, 190, 200, 30);
l6.setBounds(80, 230, 200, 30);
l7.setBounds(80, 270, 200, 30);
l8.setBounds(80, 310, 200, 30);
tf1.setBounds(300, 70, 200, 30);
tf2.setBounds(300, 110, 200, 30);
p1.setBounds(300, 150, 200, 30);
p2.setBounds(300, 190, 200, 30);
tf5.setBounds(300, 230, 200, 30);
tf6.setBounds(300, 270, 200, 30);
tf7.setBounds(300, 310, 200, 30);
btn1.setBounds(50, 350, 100, 30);
btn2.setBounds(170, 350, 100, 30);
add(l1);
add(l2);
add(tf1);
add(l3);
add(tf2);
add(l4);
add(p1);
```

```

        add(l5);
        add(p2);
        add(l6);
        add(tf5);
        add(l7);
        add(tf6);
        add(l8);
        add(tf7);
        add(btn1);
        add(btn2);
    }
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource() == btn1)
        {
            int x = 0;
            String s1 = tf1.getText();
            String s2 = tf2.getText();
            char[] s3 = p1.getPassword();
            char[] s4 = p2.getPassword();
            String s8 = new String(s3);
            String s9 = new String(s4);
            String s5 = tf5.getText();
            String s6 = tf6.getText();
            String s7 = tf7.getText();
            if (s8.equals(s9))
            {
                try
                {
                    Class.forName("oracle.jdbc.driver.OracleDriver");
                    Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@mcndesкто
p07:1521:xe", "sandeep", "welcome");

```

```

        PreparedStatement ps =
con.prepareStatement("insert into reg values(?,?,?,?,?,?)");
        ps.setString(1, s1);
        ps.setString(2, s2);
        ps.setString(3, s8);
        ps.setString(4, s5);
        ps.setString(5, s6);
        ps.setString(6, s7);
        ResultSet rs = ps.executeQuery();
        x++;
        if (x > 0)
        {
            JOptionPane.showMessageDialog(btn1, "Data
Saved Successfully");
        }
    }
    catch (Exception ex)
    {
        System.out.println(ex);
    }
}
else
{
    JOptionPane.showMessageDialog(btn1, "Password
Does Not Match");
}
}
else
{
    tf1.setText("");
    tf2.setText("");
    p1.setText("");
    p2.setText("");
}

```

```

        tf5.setText("");
        tf6.setText("");
        tf7.setText("");
    }
}
public static void main(String args[])
{
    new Registration();
}
}

```

Output:

The image displays two screenshots of a Java application window titled "Registration Form in Java". The window contains a form titled "Registration Form in Windows Form:" with the following fields and controls:

- Name:
- Email-ID:
- Create Password:
- Confirm Password:
- Country:
- State:
- Phone No:
- Submit button
- Clear button

The top screenshot shows the form with pre-filled data. The bottom screenshot shows the same form with all input fields empty, demonstrating the state after a reset or new entry.

Set C:

- 1. Write a java program to accept the details of employee employee eno,ename, sal and display it on next frame using appropriate even .**

```
import java.awt.*;
import java.awt.event.*;
class Slip24 extends Frame implements ActionListener
{
    Label l1,l2,l3,l;
    TextField txt1,txt2,txt3;
    Button submit,clear;
    Panel p1;
    Slip24()
    {
        l=new Label("EMPLOYEE INFORMTION");
        l1=new Label("Name ");
        l2=new Label("Address ");
        l3=new Label("Salary ");
        txt1=new TextField(20);
        txt2=new TextField(20);
        txt3=new TextField(20);
        submit=new Button("submit");
        submit.addActionListener(this);
        clear=new Button("Clear");
        clear.addActionListener(this);
        p1=new Panel();
        //p1.setLayout(new GridLayout(6,2));
        p1.add(l1);
        p1.add(txt1);
        p1.add(l2);
        p1.add(txt2);
        p1.add(l3);
        p1.add(txt3);
```



```

        p1.add(submit);
        p1.add(clear
        add(p1);
        setVisible(true);
        setSize(400,400);
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==submit)
        {
            new
Employee_Detail(txt1.getText(),txt2.getText(),txt3.getText(
));
        }
        if(ae.getSource()==clear)
        {
            txt1.setText("");
            txt2.setText("");
            txt3.setText("");
        }
    }
    public static void main(String args[])
    {
        new Slip24();
    }
}

```

2. Write a java program to display at least five records of employee in JTable.(Eno, Ename,Sal).

```

import javax.swing.*;
import java.awt.event.*;
import java.sql.*;
import java.util.Vector;
class Slip28_2 implements ItemListener

```

```

{
JFrame f;
JComboBox cb;
JPanel p;
JLabel l,l1,l2,l3;
Connection conn = null;
Statement stmt = null;
PreparedStatement ps = null;
ResultSet rs,rs1 = null;
Slip28_2()
{
f = new JFrame("Employee Information");
p = new JPanel();
l = new JLabel("Select emp No : ");
l1 = new JLabel();
l2 = new JLabel();
l3 = new JLabel();
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
conn = DriverManager.getConnection("jdbc:odbc:dsn11");
stmt = conn.createStatement();
Vector v = new Vector();
rs = stmt.executeQuery("select eno from employee");
while(rs.next())
{
v.add(rs.getInt(1));
}
cb = new JComboBox(v);
cb.addItemListener(this);
}
catch(Exception e)
{

```

```
e.printStackTrace();
}
p.add(l);
p.add(cb);
p.add(l1);
p.add(l2);
p.add(l3);
f.add(p);
f.setSize(400, 400);
f.setVisible(true);
}
public void itemStateChanged(ItemEvent e)
{
JComboBox source = (JComboBox)e.getSource();
Integer emp_no = (Integer)source.getSelectedItem();
try
{
rs1 = stmt.executeQuery("select * from employee where
eno="+emp_no);
if(rs1.next())
{
l1.setText("\nEmp No : "+rs1.getInt(1));
l2.setText("\nName : "+rs1.getString(2));
l3.setText("\nSalary : "+rs1.getInt(3));
}

}
catch(Exception ex)
{
System.out.println(ex);
}
}
public static void main(String args[])
```

```
{
    Slip28_2 obj = new Slip28_2();
}
}
```

3. Write a java Program to change the color of frame. If user clicks on close button then the position of frame should get change.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

```
public class ColorChanger extends JFrame {
    private JButton closeButton;
```

```
    public ColorChanger() {
        super("Color Changer");
```

```
        // Set the background color to random values
        getContentPane().setBackground(new
        Color((int)(Math.random() * 256), (int)(Math.random() * 256),
        (int)(Math.random() * 256)));
```

```
        // Create the close button
        closeButton = new JButton("Close");
        closeButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Change the position of the frame
                setLocation((int)(Math.random() * 500),
                (int)(Math.random() * 500));
            }
        });
        add(closeButton, BorderLayout.SOUTH);
```

```
setSize(300, 200);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);
}
```

```
public static void main(String[] args) {
    ColorChanger cc = new ColorChanger();
    cc.setVisible(true);
}
}
```

4. Write a java program to display following screen.
Create a Compound Interest Calculator.

```
import java.awt.*;
import java.awt.event.*;
import java.text.DecimalFormat;
import javax.swing.*;
```

```
public class CompoundInterestCalculator extends JFrame {
    private JTextField principalField;
    private JTextField rateField;
    private JTextField yearsField;
    private JTextField resultField;
    private JButton calculateButton;
```

```
public CompoundInterestCalculator() {
    super("Compound Interest Calculator");
```

```
// Create the label and text fields
JLabel principalLabel = new JLabel("Principal: ");
principalField = new JTextField(10);
JLabel rateLabel = new JLabel("Rate: ");
rateField = new JTextField(10);
JLabel yearsLabel = new JLabel("Years: ");
```

```

yearsField = new JTextField(10);
JLabel resultLabel = new JLabel("Result: ");
resultField = new JTextField(10);
resultField.setEditable(false);

// Create the calculate button
calculateButton = new JButton("Calculate");
calculateButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Calculate the compound interest
        double principal =
Double.parseDouble(principalField.getText());
        double rate = Double.parseDouble(rateField.getText()) /
100;
        int years = Integer.parseInt(yearsField.getText());
        double result = principal * Math.pow(1 + rate, years);

        // Display the result
        DecimalFormat df = new DecimalFormat("#.00");
        resultField.setText(df.format(result));
    }
});

// Add the components to the frame
JPanel panel = new JPanel();
panel.add(principalLabel);
panel.add(principalField);
panel.add(rateLabel);
panel.add(rateField);
panel.add(yearsLabel);
panel.add(yearsField);
panel.add(resultLabel);
panel.add(resultField);

```

```

    panel.add(calculateButton);
    add(panel, BorderLayout.CENTER);

    setSize(300, 150);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
}

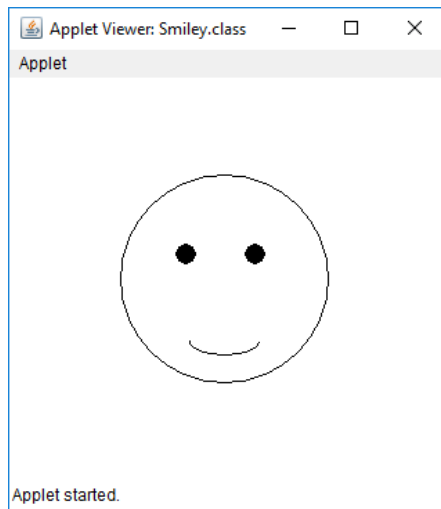
public static void main(String[] args) {
    CompoundInterestCalculator cic = new
CompoundInterestCalculator();
    cic.setVisible(true);
}
}

5. Write an applet application to display smiley and sad face.
import java.awt.*;
import java.applet.*;
/*
<applet code= "smileface.java" height="600" width="600">
</applet>
*/
public class smileface extends java.applet.Applet implements
Runnable {
    int aflag;
    Thread t;
    public void init() {
        t=new Thread(this); aflag=0;
        t.start();
    }
    public void run()
    {
        try
        {

```

```
if (aflag==0)
{
    t.sleep(1000);
    aflag=1;
} else
{
    t.sleep(1000);
    aflag=0;
}
repaint();
run();
}
catch(Exception e)
{
}
}

public void paint(Graphics g) {
    g.drawOval(100,100,100,100);
    g.fillOval(120,125,20,20);
    g.fillOval(160,125,20,20);
    g.drawLine(150,135,150,165);
    if (aflag==0)
    { g.drawArc(140,160,20,20,0,-180);
      aflag=1;
    }
    else
    {
        g.drawArc(140,160,20,20,0,180);
        aflag=0;
    }
}
}
```

Assignment 1: Introduction to Basic Python

Practice Set:

1. A cashier has currency notes of denomination 1, 5 and 10. Write python script to accept the amount to be withdrawn from the user and print the total number of currency notes of each denomination the cashier will have to give.

```
# define the denominations of currency notes
denominations = [1, 5, 10]
```

```
# prompt the user for the amount to be withdrawn
amount = int(input('Enter the amount to be
withdrawn: '))
```

```
# initialize the count of each denomination to 0
count = [0, 0, 0]
```

```
# calculate the number of notes for each denomination
for i, d in enumerate(denominations):
    count[i] = amount // d
    amount = amount % d
```

```
# print the total number of notes for each
denomination
for i, d in enumerate(denominations):
    print(f'Number of {d} notes: {count[i]}')
```

OUTPUT

Enter the amount to be withdrawn: 27

Number of 1 notes: 2

Number of 5 notes: 0

Number of 10 notes: 2

2. Write a python script to accepts annual basic salary of an employee and calculates and displays the Income tax as per the following rules.

Basic: < 2,50,000 Tax = 0

Basic: 2,50,000 to 5,00,000 Tax = 10%

Basic: > 5,00,000 Tax = 20

prompt the user for the annual basic salary

```
annual_basic_salary = int(input('Enter the annual basic salary: '))
```

calculate the income tax based on the rules provided

```
if annual_basic_salary < 250000:
```

```
    tax = 0
```

```
elif annual_basic_salary <= 500000:
```

```
    tax = 0.1 * annual_basic_salary
```

```
else:
```

```
    tax = 0.2 * annual_basic_salary
```

print the income tax

```
print(f'Income tax: {tax}')
```

OUTPUT:

Enter the annual basic salary: 600000

Income tax: 120000.0

3. Write python script to accept the x and y coordinate of a point and find the quadrant in which the point lies.

```
# prompt the user for the x and y coordinates of the point
```

```
x = int(input('Enter the x coordinate of the point: '))
```

```
y = int(input('Enter the y coordinate of the point: '))
```

```
# determine the quadrant in which the point lies
```

```
if x > 0 and y > 0:
```

```
    quadrant = 1
```

```
elif x < 0 and y > 0:
```

```
    quadrant = 2
```

```
elif x < 0 and y < 0:
```

```
    quadrant = 3
```

```
elif x > 0 and y < 0:
```

```
    quadrant = 4
```

```
else:
```

```
    quadrant = 0
```

```
# print the quadrant
```

```
print(f'Quadrant: {quadrant}')
```

OUTPUT:

Enter the x coordinate of the point: -3

Enter the y coordinate of the point: 4

Quadrant: 2

4. Write a python script to accept the cost price and selling price from the keyboard. Find out if the seller has made a profit or loss and display how much profit or loss has been made.

prompt the user for the cost price and selling price

cost_price = float(input('Enter the cost price: '))

selling_price = float(input('Enter the selling price: '))

calculate the profit or loss

profit_or_loss = selling_price - cost_price

determine if the seller has made a profit or loss

if profit_or_loss > 0:

print(f'The seller has made a profit of {profit_or_loss}')

elif profit_or_loss < 0:

print(f'The seller has made a loss of {abs(profit_or_loss)}')

else:

print('The seller has neither made a profit nor a loss')

OUTPUT:

Enter the cost price: 100

Enter the selling price: 150

The seller has made a profit of 50.0

Set A:

- 1. Write python script to calculate sum of digits of a given input number.**

prompt the user for the input number

number = int(input('Enter a number: '))

initialize the sum of digits to 0

sum_of_digits = 0

calculate the sum of digits

while number > 0:

digit = number % 10

sum_of_digits += digit

number = number // 10

print the sum of digits

print(f'Sum of digits: {sum_of_digits}')

OUTPUT :

Enter a number: 12345

Sum of digits: 15

- 2. Write python script to check whether a input number is Armstrong number or not.**

prompt the user for the input number

number = int(input('Enter a number: '))

```
# initialize the sum of digits to 0
sum_of_digits = 0
```

```
# calculate the sum of digits
temp = number
while temp > 0:
    digit = temp % 10
    sum_of_digits += digit ** 3
    temp = temp // 10
```

```
# check if the input number is an Armstrong number
if number == sum_of_digits:
    print(f'{number} is an Armstrong number')
else:
    print(f'{number} is not an Armstrong number')
```

OUTPUT :

```
Enter a number: 153
153 is an Armstrong number
```

3. Write python script to check whether a input number is perfect number or not.

```
# prompt the user for the input number
number = int(input('Enter a number: '))
```

```
# initialize the sum of factors to 0
sum_of_factors = 0
```

```
# calculate the sum of factors
```

```
for i in range(1, number):
```

```
    if number % i == 0:
```

```
        sum_of_factors += i
```

```
# check if the input number is a perfect number
```

```
if number == sum_of_factors:
```

```
    print(f'{number} is a perfect number')
```

```
else:
```

```
    print(f'{number} is not a perfect number')
```

OUTPUT :

Enter a number: 6

6 is a perfect number

4. Write a program to calculate xy

```
x = int(input("Enter the value of X: "))
```

```
y = int(input("Enter the value of Y: "))
```

```
print(pow(x, y))
```


4. Write a program to check whether a input number is palindrome or not.

prompt the user for the input number

number = input('Enter a number: ')

check if the input number is a palindrome

if number == number[::-1]:

print(f'{number} is a palindrome')

else:

print(f'{number} is not a palindrome')

OUTPUT :

Enter a number: 121

121 is a palindrome

5. Write a program to calculate sum of first and last digit of a number.

prompt the user for the input number

number = int(input('Enter a number: '))

calculate the first and last digits

first_digit = number // (10 ** (len(str(number)) - 1))

last_digit = number % 10

calculate the sum of the first and last digits

sum_of_digits = first_digit + last_digit

print the result

print(f'Sum of first and last digits: {sum_of_digits}')

OUTPUT :

Enter a number: 12345

Sum of first and last digits: 6

Set B:

1. Write a program to accept a number and count number of even, odd, zero digits within that number.

prompt the user for the input number

number = input('Enter a number: ')

```
# initialize the counters for each digit type
```

```
even_count = 0
```

```
odd_count = 0
```

```
zero_count = 0
```

```
# count the number of each digit type
```

```
for digit in number:
```

```
    if digit == '0':
```

```
        zero_count += 1
```

```
    elif int(digit) % 2 == 0:
```

```
        even_count += 1
```

```
    else:
```

```
        odd_count += 1
```

```
# print the counts
```

```
print(f'Number of even digits: {even_count}')
```

```
print(f'Number of odd digits: {odd_count}')
```

```
print(f'Number of zero digits: {zero_count}')
```

```
OUTPUT :
```

```
Enter a number: 123004567
```

```
Number of even digits: 5
```

```
Number of odd digits: 2
```

```
Number of zero digits: 3
```

- 2. Write a program to accept a binary number and convert it into decimal number.**

```
# prompt the user for the binary number
```

```
binary = input('Enter a binary number: ')
```

```
# initialize the decimal number to 0
```

```
decimal = 0
```

```
# convert the binary number to decimal
```

```
for i, digit in enumerate(reversed(binary)):
```

```
    decimal += int(digit) * (2 ** i)
```

```
# print the decimal number
```

```
print(f'Decimal equivalent: {decimal}')
```

```
OUTPUT :
```

```
Enter a binary number: 1101
```

```
Decimal equivalent: 13
```

3. Write a program which accepts an integer value as command line and print “Ok” if value is between 1 to 50 (both inclusive) otherwise it prints ”Out of range”

```
import sys
```

```
# check if a command line argument was provided
```

```
if len(sys.argv) < 2:
```

```
    print('Please provide an integer value as a command line argument.')
```

```
    sys.exit(1)
```

```
# get the integer value from the command line argument
```

```
number = int(sys.argv[1])
```

```
# check if the number is within the range 1 to 50
```

```
if 1 <= number <= 50:
```

```
    print('Ok')
```

```
else:
```

```
    print('Out of range')
```

OUTPUT :

```
$ python3 program.py 25
```

Ok

```
$ python3 program.py 55
```

Out of range

4. Write a program which accept an integer value 'n' and display all prime numbers till 'n'.

```
# prompt the user for the value of n
```

```
n = int(input('Enter a value for n: '))
```

```
# print all prime numbers till n
```

```
print('Prime numbers till n:')
```

```
for number in range(2, n+1):
```

```
    is_prime = True
```

```
for i in range(2, number):  
    if number % i == 0:  
        is_prime = False  
        break  
if is_prime:  
    print(number)
```

OUTPUT :

Enter a value for n: 10

Prime numbers till n:

2

3

5

7

5. Write python script to accept two numbers as range and display multiplication table of all numbers within that range.

prompt the user for the range

start = int(input('Enter the start of the range: '))

end = int(input('Enter the end of the range: '))

```
# print the multiplication table
print('Multiplication table:')
for number in range(start, end+1):
    for i in range(1, 11):
        print(f'{number} x {i} = {number * i}')
    print()
```

OUTPUT :

Enter the start of the range: 5

Enter the end of the range: 8

Multiplication table:

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

$$6 \times 1 = 6$$

$$6 \times 2 = 12$$

$$6 \times 3 = 18$$

$$6 \times 4 = 24$$

$$6 \times 5 = 30$$

$$6 \times 6 = 36$$

$$6 \times 7 = 42$$

$$6 \times 8 = 48$$

$$6 \times 9 = 54$$

$$6 \times 10 = 60$$

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

$$7 \times 4 = 28$$

Set C:

1. Write a python script to generate the following pattern upto n lines

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

```
# prompt the user for the number of lines
n = int(input('Enter the number of lines: '))
```

```
# generate the pattern
for i in range(1, n+1):
    for j in range(1, i+1):
        print(j, end=' ')
    for j in range(i-1, 0, -1):
        print(j, end=' ')
    print()
```

OUTPUT :

Enter the number of lines: 4

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

Assignment 2: Working with String and List

Assignments:

Practice Set

- 1. Write a python script to create a list and display the list element in reverse order.**

```
# create a list of numbers
```

```
numbers = [1, 2, 3, 4, 5]
```

```
# use the reversed() function to reverse the order of  
the list
```

```
reversed_numbers = reversed(numbers)
```

```
# print the reversed list
```

```
print(list(reversed_numbers))
```

```
OUTPUT : 5 4 3 2 1
```

- 2. Write a python script to display alternate characters of string from both the direction.**

```
# define a string
```

```
my_string = "Hello World!"
```

```
# get the length of the string
```

```
string_length = len(my_string)
```

```
# initialize an empty string for the result
```

```
result = ""
```

```
# iterate over the string from the start and end
```

```
for i in range(string_length):
```

add the character at the current index from the start of the string

result += my_string[i]

add the character at the current index from the end of the string

we need to subtract 2 from the string length because we already added

one character from the start of the string and the index starts at 0

result += my_string[string_length - i - 2]

print the result

print(result)

OUTPUT :

The output of this script will be HloolWlelrd!.

3. Write a python program to count vowels and consonants in a string.

define a string

my_string = "Hello World!"

initialize counters for vowels and consonants

vowels = 0

consonants = 0

iterate over the characters in the string

for char in my_string:

check if the current character is a vowel

```
if char.lower() in "aeiou":  
    vowels += 1  
# check if the current character is a consonant  
elif char.isalpha():  
    consonants += 1  
  
# print the results  
print(f"Vowels: {vowels}")  
print(f"Consonants: {consonants}")
```

OUTPUT :

Vowels: 3

Consonants: 7

Set A

1. Write a python script which accepts 5 integer values and prints “DUPLICATES” if any of the values entered are duplicates otherwise it prints “ALL UNIQUE”. Example: Let 5 integers are (32, 45, 90, 45, 6) then output “DUPLICATES” to be printed.

```
# define a string  
my_string = "Hello World!"
```

```
# initialize counters for vowels and consonants  
vowels = 0  
consonants = 0
```

```
# iterate over the characters in the string
```

```
for char in my_string:
    # check if the current character is a vowel
    if char.lower() in "aeiou":
        vowels += 1
    # check if the current character is a consonant
    elif char.isalpha():
        consonants += 1

# print the results
print(f"Vowels: {vowels}")
print(f"Consonants: {consonants}")
```

OUTPUT :

The output of this script will be DUPLICATES.

2. Write a python script to count the number of characters (character frequency) in a string. Sample String : google.com'. Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}

define a string

my_string = "google.com"

initialize a dictionary to store the character frequencies

char_freq = {}

iterate over the characters in the string

for char in my_string:

```
# check if the current character is already in the
dictionary
if char in char_freq:
    # if it is, increment its count
    char_freq[char] += 1
else:
    # if it is not, add it to the dictionary and set its
count to 1
    char_freq[char] = 1

# print the character frequencies
print(char_freq)
```

The output of this script will be {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}.

3. Write a Python program to remove the characters which have odd index values of a given string.

```
# define a string
my_string = "Hello World!"

# initialize an empty string for the result
result = ""

# iterate over the characters in the string
for i, char in enumerate(my_string):
    # check if the index of the current character is even
    if i % 2 == 0:
        # if it is, add it to the result
```

```
result += char
```

```
# print the result  
print(result)
```

The output of this script will be Hlo ol!.

4. Write a program to implement the concept of stack using list.

```
stack = []
```

```
stack.append('a')  
stack.append('b')  
stack.append('c')
```

```
print('Initial stack')  
print(stack)
```

```
print('\nElements popped from stack:')  
print(stack.pop())  
print(stack.pop())  
print(stack.pop())
```

```
print('\nStack after elements are popped:')  
print(stack)
```

OUTPUT :

Initial stack

['a', 'b', 'c']

Elements popped from stack:

c

b

a

5. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '\$', except the first char itself.

Sample String: 'restart' Expected Result :

'resta\$t'

```
str = input("Enter a string: ")
```

```
char = str[0]
```

```
str = str.replace(char, '$')
```

```
str = char + str[1:]
```

```
print(str)
```

OUTPUT :

Hello

\$ello

SET B :

1. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Sample String : 'General12'

Expected Result : 'Ge12'

Sample String : 'Ka'

Expected Result : 'KaKa'

Sample String : ' K'

```
str = input("Enter a string: ")
```

```
if len(str) >= 2:
```

```
    expected_str = str[0:3] + str[-3:]
```

```
else:
```

```
    expected_str = "Empty String"
```

```
print(expected_str)
```

2. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Sample String : 'abc', 'xyz'

Expected Result : 'xycabz'

```
def swap_strings(str1, str2):
```

```
# Swap the first two characters of each string
str1_swapped = str2[:2] + str1[2:]
str2_swapped = str1[:2] + str2[2:]

# Concatenate the swapped strings with a space in
between
return str1_swapped + ' ' + str2_swapped

# Test the swap_strings() function
print(swap_strings('abc', 'xyz'))
# Expected output: 'xycabz'
```

3. Write a Python program to count the occurrences of each word in a given sentence.

```
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts

str = input("Enter a sentence: ")

print(word_count(str))
```

4. Write a program to implement the concept of queue using list.

```
queue = []
```

```
queue.append('a')
```

```
queue.append('b')
```

```
queue.append('c')
```

```
print("Initial queue")
```

```
print(queue)
```

```
print("\nElements dequeued from queue")
```

```
print(queue.pop(0))
```

```
print(queue.pop(0))
```

```
print(queue.pop(0))
```

```
print("\nQueue after removing elements")
```

```
print(queue)
```

5. Write a python program to count repeated characters in a string.

Sample string:

'thequickbrownfoxjumpsoverthelazydog'

Expected output:

o 4

e 3

u 2

h 2

r 2

t 2

```
str = input("Enter a string: ")
```

```
char_count = {}
```

```
for char in str:
```

```
    if char in char_count:
```

```
        char_count[char] += 1
```

```
    else:
```

```
        char_count[char] = 1
```

```
print(char_count)
```

Set C:

1. Write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

```
def binarySearch(arr, l, r, ele):  
    if r >= l:  
        mid = l + (r - l) // 2  
  
        if arr[mid] == ele:  
            return mid  
        elif arr[mid] > ele:  
            return binarySearch(arr, l, mid - 1, ele)  
        else:  
            return binarySearch(arr, mid + 1, r, ele)  
    else:  
        return -1
```

```
arr = [10, 20, 30, 40, 50]
```

```
ele = 40
```

```
result = binarySearch(arr, 0, len(arr)-1, ele)
```

if result != -1:

print(f"Element {ele} is present at index {result}.")

else:

print("Element is not present in array")

Assignment 3: Working With Tuples, Sets and Dictionaries

Assignments:

Set A:

1. Write a Python program to find maximum and the minimum value in a set.

define a set of numbers

numbers = {1, 2, 3, 4, 5}

find the maximum value in the set

max_value = max(numbers)

find the minimum value in the set

min_value = min(numbers)

print the results

print("The maximum value in the set is:", max_value)

print("The minimum value in the set is:", min_value)

2. Write a Python program to add an item in a tuple.

#create a tuple

tuplex = (4, 6, 2, 8, 3, 1)

print(tuplex)

#tuples are immutable, so you can not add new elements

#using merge of tuples with the + operator you can add an element and it will create a new tuple

```
tuplex = tuplex + (9,)
print(tuplex)

#adding items in a specific index
tuplex = tuplex[:5] + (15, 20, 25) + tuplex[:5]
print(tuplex)

#converting the tuple to list
listx = list(tuplex)

#use different ways to add items in list
listx.append(30)

tuplex = tuple(listx)
print(tuplex)
```

OUTPUT :

```
(4, 6, 2, 8, 3, 1)
(4, 6, 2, 8, 3, 1, 9)
(4, 6, 2, 8, 3, 15, 20, 25, 4, 6, 2, 8, 3)
(4, 6, 2, 8, 3, 15, 20, 25, 4, 6, 2, 8, 3, 30)
```

3. Write a Python program to convert a tuple to a string.

```
# define a tuple
tuple = ("a", "b", "c", "d")

# convert the tuple to a string
string = "".join(tuple)

# print the result
```



```
print("The tuple converted to a string is:", string)
```

```
# define a tuple
```

```
tuple = ("a", "b", "c", "d")
```

```
# convert the tuple to a string
```

```
string = ",".join(tuple)
```

```
# print the result
```

```
print("The tuple converted to a string is:", string)
```

OUTPUT :

The tuple converted to a string is: a,b,c,d

4. Write a Python program to create an intersection of sets.

```
setx = set(["green", "blue"])
```

```
sety = set(["blue", "yellow"])
```

```
print("Original set elements:")
```

```
print(setx)
```

```
print(sety)
```

```
print("\nIntersection of two said sets:")
```

```
setz = setx & sety
```

```
print(setz)
```

output :

Original set elements:

```
{'green', 'blue'}
```

```
{'blue', 'yellow'}
```

Intersection of two said sets:

```
{'blue'}
```

5. Write a Python program to create a union of sets.

```
# create two sets
```

```
set1 = {1, 2, 3}
```

```
set2 = {3, 4, 5}
```

```
# create the union of the sets
```

```
union = set1.union(set2)
```

```
# print the union
```

```
print(union)
```

```
{1, 2, 3, 4, 5}
```

6. Write a Python script to check if a given key already exists in a dictionary.

```
# define a dictionary
```

```
my_dict = {
```

```
    "key1": "value1",
```

```
    "key2": "value2",
```

```
    "key3": "value3"
```

```
}
```

```
# define the key to check
```

```
key = "key1"
```

```
# check if the key exists in the dictionary
```

```
if key in my_dict:
```

```
    print(f"The key '{key}' exists in the dictionary.")
```

```
else:
```

```
    print(f"The key '{key}' does not exist in the dictionary.")
```

output :

7. Write a Python script to sort (ascending and descending) a dictionary by value.

```
# define a dictionary
```

```
my_dict = {
```

```
    "key1": 10,
```

```
    "key2": 5,
```

```
    "key3": 8
```

```
}
```

```
# create a dictionary view object
```

```
dict_view = my_dict.items()
```

```
# sort the dictionary view object in ascending order by value
sorted_dict_asc = sorted(dict_view, key=lambda item: item[1])
```

```
# sort the dictionary view object in descending order by value
sorted_dict_desc = sorted(dict_view, key=lambda item: item[1],
reverse=True)
```

```
# print the sorted dictionaries
print("Ascending order:", sorted_dict_asc)
print("Descending order:", sorted_dict_desc)
```

Ascending order: [('key2', 5), ('key3', 8), ('key1', 10)]

Descending order: [('key1', 10), ('key3', 8), ('key2', 5)]

Set B:

1. Write a Python program to create set difference and a symmetric difference.

```
# create two sets
```

```
set1 = {1, 2, 3}
```

```
set2 = {3, 4, 5}
```

```
# create the set difference of the sets
```

```
difference = set1.difference(set2)
```

print the set difference

print("Set difference:", difference)

create the symmetric difference of the sets

symmetric_difference = set1.symmetric_difference(set2)

print the symmetric difference

print("Symmetric difference:", symmetric_difference)

output :

Set difference: {1, 2}

Symmetric difference: {1, 2, 4, 5}

2. Write a Python program to create a list of tuples with the first element as the number and second element as the square of the number.

create a list of numbers

numbers = [1, 2, 3, 4, 5]

create a list of tuples

tuples = [(num, num2) for num in numbers]**

print the list of tuples

print(tuples)

output :

[(1, 1), (2, 4), (3, 9), (4, 16), (5, 25)]

3. Write a Python program to unpack a tuple in several variables.

define a tuple

my_tuple = (1, 2, 3)

unpack the tuple into several variables

a, b, c = my_tuple

print the variables

print("a:", a)

print("b:", b)

print("c:", c)

output :

a: 1

b: 2

c: 3

ValueError: too many values to unpack (expected 2)

4. Write a Python program to get the 4th element from front and 4th element from last of a tuple.

define a tuple

my_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

get the 4th element from the front

fourth_front = my_tuple[3]

get the 4th element from the last

fourth_last = my_tuple[-4]

print the elements

print("Fourth element from front:", fourth_front)

print("Fourth element from last:", fourth_last)

Output :

Fourth element from front: 4

Fourth element from last: 7

5. Write a Python program to find the repeated items of a tuple.

define a tuple

my_tuple = (1, 2, 3, 4, 5, 5, 6, 7, 8, 9, 10)

create a set from the tuple

my_set = set(my_tuple)

find the repeated items by subtracting the set from the tuple

repeated_items = [item for item in my_tuple if item not in my_set]

print the repeated items

print("Repeated items:", repeated_items)

output :

Repeated items: []

6. Write a Python program to check whether an element exists within a tuple.

define a tuple

my_tuple = (1, 2, 3, 4, 5)

define the element to check

element = 4

check if the element exists in the tuple

if element in my_tuple:

print(f"The element '{element}' exists in the tuple.")

else:

print(f"The element '{element}' does not exist in the tuple.")

output :

The element '4' exists in the tuple.

7. Write a Python script to concatenate following dictionaries to create a new one. Sample

Dictionary : dic1={1:10, 2:20} dic2={3:30, 4:40} dic3={5:50,6:60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

define the dictionaries to concatenate

dic1 = {1:10, 2:20}

dic2 = {3:30, 4:40}

dic3 = {5:50, 6:60}

concatenate the dictionaries

dic4 = {dic1, **dic2, **dic3}**

print the concatenated dictionary

print(dic4)

output :

{1: 10, 2: 50, 3: 40, 4: 60}

Set C:

- 1. Write a Python program to create a shallow copy of sets.**

define a set

my_set = {1, 2, 3, 4, 5}

create a shallow copy of the set

my_set_copy = my_set.copy()

print the shallow copy

print("Shallow copy:", my_set_copy)

output :

Shallow copy: {1, 2, 3, 4, 5}

- 2. Write a Python program to combine two dictionary adding values for common keys.**

d1 = {'a': 100, 'b': 200, 'c': 300}

d2 = {'a': 300, 'b': 200, 'd': 400}

Sample output: Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})

from collections import Counter

```
d1 = {'a': 100, 'b': 200, 'c':300}
```

```
d2 = {'a': 300, 'b': 200, 'd':400}
```

```
# Combine the dictionaries
```

```
d = Counter(d1) + Counter(d2)
```

```
# Print the resulting dictionary
```

```
print(d)
```

Output :

```
Counter({'a': 400, 'b': 400, 'd': 400, 'c': 300})
```

Assignment 4: Working with Functions, Modules and Packages

Set A:

1. Write a recursive function which print string in reverse order.

```
def print_reverse(s: str) -> None:
```

```
    # Base case: if the string is empty, do nothing
```

```
    if s == '':
```

```
        return
```

```
    # Recursive case: print the last character of the string,
```

```
    # then call the function again with the remaining characters
```

```
print(s[-1], end='')  
print_reverse(s[:-1])
```

output :

```
print_reverse('hello')  
olleh
```

2. Write a python script using function to calculate X^Y

```
def power(x: int, y: int) -> int:
```

```
    # If Y is 0, return 1
```

```
    if y == 0:
```

```
        return 1
```

```
    # If Y is negative, return 1 divided by the power of X to the  
    absolute value of Y
```

```
    if y < 0:
```

```
        return 1 / power(x, abs(y))
```

```
    # If Y is odd, return X times the power of X to Y-1
```

```
    if y % 2 == 1:
```

```
        return x * power(x, y - 1)
```

```
    # If Y is even, return the square of the power of X to Y/2
```

```
    return power(x, y // 2) ** 2
```

output :

```
x = 2
```

y = 8

result = power(x, y)

print(f'{x}^{y} = {result}')

output :

2^8 = 256

3. Define a function that accept two strings as input and find union and intersection of them.

def union_intersection(s1: str, s2: str) -> tuple:

Find the union of the two strings by converting them to sets

and taking the union of the sets

union = set(s1) | set(s2)

Find the intersection of the two strings by converting them to sets

and taking the intersection of the sets

intersection = set(s1) & set(s2)

Return the union and intersection as a tuple

return union, intersection

output :

Union: {'d', 'l', 'r', 'o', 'w', 'h', 'e'}

Intersection: set()

4. Write a recursive function to calculate sum of digits of a given input number.

```
def sum_digits(n: int) -> int:
```

```
    # Base case: if the number is 0, return 0
```

```
    if n == 0:
```

```
        return 0
```

```
    # Recursive case: add the last digit of the number to the sum of  
    the remaining digits
```

```
    return n % 10 + sum_digits(n // 10)
```

output :

```
n = 12345
```

```
result = sum_digits(n)
```

```
print(f'The sum of the digits of {n} is {result}')
```

The sum of the digits of 12345 is 15

5. Write generator function which generate even numbers up to n.

```
def even_numbers(n: int) -> int:
```

```
    # Initialize a variable to track the current number
```

```
    current = 0
```

```
    # Keep generating numbers until we reach or exceed n
```

```
    while current <= n:
```

```
        # Yield the current number if it is even
```

```
    if current % 2 == 0:
        yield current

    # Increment the current number
    current += 1

# Print the first 10 even numbers
for i in even_numbers(20):
    print(i)
```

output :

0
2
4
6
8
10
12
14
16
18
20

Set B:

1. Write a python script to generate Fibonacci terms using generator function.

```
def fibonacci() -> int:
```

Initialize the first two terms of the sequence

a, b = 0, 1

Keep generating terms until the function is terminated

while True:

Yield the current term

yield a

Calculate the next term and update the current terms

a, b = b, a + b

Output :

0

1

1

2

3

5

8

13

21

34

2. Write python script using package to calculate area and volume of cylinder and cuboids.

import math

Define the radius and height of the cylinder

cylinder_radius = 5

cylinder_height = 10

Calculate the area of the cylinder

cylinder_area = 2 * math.pi * cylinder_radius2 + 2 * math.pi *
cylinder_radius * cylinder_height**

Calculate the volume of the cylinder

cylinder_volume = math.pi * cylinder_radius2 * cylinder_height**

Define the length, width, and height of the cuboid

cuboid_length = 10

cuboid_width = 5

cuboid_height = 20

Calculate the area of the cuboid

**cuboid_area = 2 * (cuboid_length * cuboid_width + cuboid_length *
cuboid_height + cuboid_width * cuboid_height)**

Calculate the volume of the cuboid

cuboid_volume = cuboid_length * cuboid_width * cuboid_height

Print the results

print(f'The area of the cylinder is {cylinder_area}')

print(f'The volume of the cylinder is {cylinder_volume}')

print(f'The area of the cuboid is {cuboid_area}')

print(f'The volume of the cuboid is {cuboid_volume}')

OUTPUT :

The area of the cylinder is 471.2388980384689

The volume of the cylinder is 314.1592653589793

The area of the cuboid is 500

The volume of the cuboid is 1000

3. Write a python script to accept decimal number and convert it to binary and octal number using function.

Define a decimal number

decimal = 15

Convert the decimal number to binary and octal

binary = bin(decimal)

octal = oct(decimal)

Print the results

print(f'{decimal} in binary is {binary}')

print(f'{decimal} in octal is {octal}')

OUTPUT :

15 in binary is 0b1111

15 in octal is 0o17

4. Write a function which print a dictionary where the keys are numbers between 1 and 20.

```
def print_dict():  
    # Create an empty dictionary  
    d = {}  
  
    # Add keys and values to the dictionary  
    for i in range(1, 21):  
        d[i] = i**2  
  
    # Print the dictionary  
    print(d)
```

OUTPUT :

```
print_dict()  
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121,  
12: 144, 13: 169, 14: 196, 15: 225, 16: 256, 17: 289, 18: 324, 19: 361,  
20: 400}
```

5. Write a generator function which generates prime numbers up to n.

```
def prime_numbers(n):  
    # Check if n is a prime number  
    def is_prime(num):  
        # Handle the special case of num being 2
```

```
if num == 2:  
    return True
```

```
# Check if num is even and not 2  
if num % 2 == 0 or num <= 1:  
    return False
```

```
i = 3  
while i * i <= num:  
    if num % i == 0:  
        return False  
    i += 2
```

```
return True
```

```
# Generate prime numbers up to n  
for i in range(2, n + 1):  
    if is_prime(i):  
        yield i
```

OUTPUT :

```
# Generate prime numbers up to 10  
for prime in prime_numbers(10):  
    print(prime)
```

Output: 2 3 5 7

Set C:

- 1. Write a program to illustrate function duck typing.**

Define a function that takes an object and uses its `quack` method

```
def quack(obj):  
    return obj.quack()
```

Define a class for ducks

```
class Duck:  
    def quack(self):  
        return "Quack!"
```

Define a class for dogs

```
class Dog:  
    def quack(self):  
        return "Woof!"
```

Create an instance of a duck and a dog

```
duck = Duck()  
dog = Dog()
```

Call the `quack` function with the duck and dog

```
print(quack(duck)) # Output: "Quack!"  
print(quack(dog)) # Output: "Woof!"
```

Assignment 5: Python Classes and Objects

Set A:

1) Write a Python Program to Accept, Delete and Display students details such as Roll.No, Name, Marks in three subject, using Classes. Also display percentage of each student.

Define a Student class

class Student:

Initialize a new Student instance

def __init__(self, roll, name, marks):

self.roll = roll

self.name = name

self.marks = marks

Calculate and return the percentage of marks

def get_percentage(self):

total_marks = sum(self.marks)

return total_marks / 3

Delete the marks for a student

def delete_marks(self):

self.marks = []

Display the details of a student

def display_details(self):

print("Roll:", self.roll)

print("Name:", self.name)

print("Marks:", self.marks)

print("Percentage:", self.get_percentage())

Create a new Student instance

student = Student(1, "John", [90, 80, 70])

```
# Display the details of the student
student.display_details()
```

```
# Delete the marks of the student
student.delete_marks()
```

```
# Display the details of the student again
student.display_details()
```

2) Write a Python program that defines a class named circle with attributes radius and center, where center is a point object and radius is number. Accept center and radius from user. Instantiate a circle object that represents a circle with its center and radius as accepted input.

```
# Define a Point class
```

```
class Point:
```

```
    # Initialize a new Point instance
```

```
    def __init__(self, x, y):
```

```
        self.x = x
```

```
        self.y = y
```

```
# Define a Circle class
```

```
class Circle:
```

```
    # Initialize a new Circle instance
```

```
    def __init__(self, center, radius):
```

```
        self.center = center
```

```
        self.radius = radius
```

```
# Accept the center and radius of the circle from the user
```

```
print("Enter the center of the circle:")
```

```
x = float(input("x: "))
```

```
y = float(input("y: "))
center = Point(x, y)
```

```
radius = float(input("Enter the radius of the circle: "))
```

```
# Instantiate a Circle object
circle = Circle(center, radius)
```

3) Write a Python class which has two methods `get_String` and `print_String`. `get_String` accept a string from the user and `print_String` print the string in upper case. Further modify the program to reverse a string word by word and print it in lower case.

```
# Define the StringPrinter class
```

```
class StringPrinter:
```

```
    # Initialize a new StringPrinter instance
```

```
    def __init__(self):
```

```
        self.string = ""
```

```
    # Accept a string from the user
```

```
    def get_string(self):
```

```
        self.string = input("Enter a string: ")
```

```
    # Print the string in upper case and reversed in lower case
```

```
    def print_string(self):
```

```
        # Print the string in upper case
```

```
        print(self.string.upper())
```

```
        # Reverse the string word by word and print it in lower
case
```

```
        words = self.string.split()
```

```
        words.reverse()
```

```
        print(" ".join(words).lower())
```

Create a new StringPrinter instance

string_printer = StringPrinter()

Accept a string from the user

string_printer.get_string()

Print the string in upper case and reversed in lower case

string_printer.print_string()

4) Write Python class to perform addition of two complex numbers using binary + operator overloading.

Define the ComplexNumber class

class ComplexNumber:

Initialize a new ComplexNumber instance

def __init__(self, real, imag):

self.real = real

self.imag = imag

Overload the binary + operator for addition of complex numbers

def __add__(self, other):

real = self.real + other.real

imag = self.imag + other.imag

return ComplexNumber(real, imag)

Create two ComplexNumber objects

c1 = ComplexNumber(1, 2)

c2 = ComplexNumber(3, 4)

Add the two complex numbers using the + operator

c3 = c1 + c2

Print the result of the addition

print(c3.real, c3.imag)

Output: 4 6

Set B:

1) Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area and volume.

Define the Rectangle class

class Rectangle:

Initialize a new Rectangle instance

def __init__(self, length, width):

self.length = length

self.width = width

Compute and return the area and volume of the rectangle

def area_and_volume(self):

area = self.length * self.width

volume = self.length * self.width * 0

return area, volume

Create a new Rectangle instance

rectangle = Rectangle(10, 20)

Compute the area and volume of the rectangle

area, volume = rectangle.area_and_volume()

Print the area and volume of the rectangle

print("Area:", area)

print("Volume:", volume)

2) Write a function named `pt_in_circle` that takes a circle and a point and returns true if point lies on the boundry of circle.

```
import math
```

```
def pt_in_circle(circle, point):
```

```
    # unpack the center and radius of the circle
```

```
    cx, cy, r = circle
```

```
    # unpack the coordinates of the point
```

```
    px, py = point
```

```
    # calculate the distance between the point and the center of  
    the circle
```

```
    d = math.sqrt((px - cx)**2 + (py - cy)**2)
```

```
    # return True if the distance is equal to the radius of the  
    circle
```

```
    return d == r
```

```
# define the circle
```

```
circle = (0, 0, 1)
```

```
# define a point on the boundary of the circle
```

```
point = (1, 0)
```

```
# check if the point lies on the boundary of the circle
```

```
result = pt_in_circle(circle, point)
```

```
print(result) # True
```

```
import math
```

```
def pt_in_circle(circle, point):
```

```
    # unpack the center and radius of the circle
```

```
    cx, cy, r = circle
```

```
    # unpack the coordinates of the point
```

```
px, py = point
```

```
# calculate the distance between the point and the center of  
the circle
```

```
d = math.sqrt((px - cx)**2 + (py - cy)**2)
```

```
# return True if the distance is less than or equal to the  
radius of the circle
```

```
return d <= r
```

3) Write a Python Program to Create a Class Set and Get All Possible Subsets from a Set of Distinct Integers.

```
class Set:
```

```
def __init__(self, elements):
```

```
# convert the input to a set to remove duplicates
```

```
self.elements = set(elements)
```

```
def get_subsets(self):
```

```
# start with the empty set
```

```
subsets = [set()]
```

```
# for each element in the set
```

```
for element in self.elements:
```

```
# for each subset in the list of subsets
```

```
for i in range(len(subsets)):
```

```
# create a new subset by adding the element to the  
original subset
```

```
new_subset = subsets[i].copy()
```

```
new_subset.add(element)
```

```
# add the new subset to the list of subsets
```

```
subsets.append(new_subset)
```

```
return subsets
```

```
# create a set
my_set = Set([1, 2, 3])

# get all possible subsets
subsets = my_set.get_subsets()

# print the list of subsets
print(subsets) # [set(), {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}]
```

4) Write a python class to accept a string and number n from user and display n repetition of strings using by overloading * operator.

```
class RepeatedString:
    def __init__(self, s, n):
        self.s = s
        self.n = n

    def __mul__(self, other):
        # if the other operand is not an integer, raise an error
        if not isinstance(other, int):
            raise TypeError('unsupported operand type(s) for *:
"RepeatedString" and "{}".format(type(other).__name__)')

        # return a new RepeatedString object with the repeated
        string and the number of repetitions
        return RepeatedString(self.s * other, self.n)

    def __str__(self):
        # return the repeated string n times
        return self.s * self.n
```

create a RepeatedString object with a string and a number of repetitions

repeated_string = RepeatedString('hello', 3)

multiply the RepeatedString object by a positive integer to repeat the string

result = repeated_string * 2 # 'hellohello'

multiply the RepeatedString object by zero to get an empty string

result = repeated_string * 0 # ''

multiply the RepeatedString object by a negative integer to get an empty string

result = repeated_string * -1 # ''

try to multiply the RepeatedString object by a non-integer value

try:

result = repeated_string * 'hello'

except TypeError as error:

print(error) # unsupported operand type(s) for *:

"RepeatedString" and "str"

Set C:

1) Python Program to Create a Class which Performs Basic Calculator Operations.

class Calculator:

def __init__(self):

self.result = 0

```
def add(self, a, b):  
    self.result = a + b  
    return self.result
```

```
def subtract(self, a, b):  
    self.result = a - b  
    return self.result
```

```
def multiply(self, a, b):  
    self.result = a * b  
    return self.result
```

```
def divide(self, a, b):  
    self.result = a / b  
    return self.result
```

```
# create a Calculator object  
calculator = Calculator()
```

```
# perform some calculations  
result = calculator.add(1, 2) # 3  
result = calculator.subtract(5, 2) # 3  
result = calculator.multiply(2, 3) # 6  
result = calculator.divide(6, 2) # 3
```

- 2) Define datetime module that provides time object. Using this module write a program that gets current date and time and print day of the week.

```
import datetime

# get the current date and time
current_time = datetime.datetime.now()

# get the current day of the week
day_of_week = current_time.strftime('%A')

# print the day of the week
print(day_of_week)
```

output :
Wednesday

Assignment 6: Inheritance

SET A:

- 1) Write a python program to demonstrate multilevel inheritance by using Base class name as "Team" which inherits Derived class name as "Dev".

class Team:

 # base class constructor

 def __init__(self, name, members):

 self.name = name

 self.members = members

 # base class method

 def show_name(self):

 print('Team name:', self.name)

 # base class method

 def show_members(self):

 print('Team members:', ', '.join(self.members))

class Dev(Team):

 # derived class constructor

 def __init__(self, name, members, language):

 # call the base class constructor

 super().__init__(name, members)

 self.language = language

 # derived class method

 def show_language(self):

 print('Programming language:', self.language)

 # create a Dev object

dev_team = Dev('Dev Team', ['Alice', 'Bob', 'Carol'],
'Python')


```
# call base class methods
dev_team.show_name() # Team name: Dev Team
dev_team.show_members() # Team members: Alice,
Bob, Carol
```

```
# call derived class method
dev_team.show_language() # Programming language:
Python
```

2) Write a python program by considering Baseclass as TeamMember and Derived class as TeamLeader use multiple inheritance concept to demonstrate the code.

```
class TeamMember:
```

```
    def __init__(self, name, team):
        self.name = name
        self.team = team
```

```
    def introduce(self):
        print(f"Hi, my name is {self.name} and I am a member
of the {self.team} team.")
```

```
class Employee:
```

```
    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
```

```
    def show_salary(self):
        print(f"{self.name} makes ${self.salary} per year.")
```

```
class TeamLeader(TeamMember, Employee):
```

```
    def __init__(self, name, team, salary):
        # Call the constructor for the TeamMember class
        TeamMember.__init__(self, name, team)
```

```
# Call the constructor for the Employee class  
Employee.__init__(self, name, salary)
```

```
# Define a new method that is specific to the TeamLeader  
class
```

```
def lead(self):  
    print(f"{self.name} is leading the {self.team} team.")
```

```
# Create an instance of the TeamLeader class  
team_leader = TeamLeader("Alice", "Marketing", 80000)
```

```
# Call methods from the TeamMember and Employee  
classes
```

```
team_leader.introduce()  
team_leader.show_salary()
```

```
# Call the lead() method, which is specific to the  
TeamLeader class
```

```
team_leader.lead()
```

**3) Write a python program to make use of issubclass () or
isinstance() functions to check the relationships of two
classes and instances.**

```
class BaseClass:  
    pass
```

```
class SubClass(BaseClass):  
    pass
```

```
# Check if SubClass is a subclass of BaseClass  
print(issubclass(SubClass, BaseClass)) # Output: True
```

```
# Create an instance of the BaseClass  
base_class_instance = BaseClass()
```

```
# Check if the base_class_instance is an instance of the
BaseClass
print(isinstance(base_class_instance, BaseClass)) # Output:
True
```

```
# Check if the base_class_instance is an instance of the
SubClass
print(isinstance(base_class_instance, SubClass)) # Output:
False
```

SET B :

- 1) Write a python program to inherit (Derived class) "course" from (base class) "University" Using hybrid inheritance concept.

```
class University:
    def __init__(self, name):
        self.name = name

    def get_name(self):
        return self.name

class Department(University):
    def __init__(self, name, department):
        University.__init__(self, name)
        self.department = department

    def get_department(self):
        return self.department

class Course(Department):
    def __init__(self, name, department, course_name):
        Department.__init__(self, name, department)
```

```
self.course_name = course_name
```

```
def get_course_name(self):  
    return self.course_name  
course = Course("MIT", "Computer Science",  
"Introduction to Artificial Intelligence")
```

```
# Get the name of the university offering the course  
print(course.get_name())  
# Output: MIT
```

```
# Get the department that the course belongs to  
print(course.get_department())  
# Output: Computer Science
```

```
# Get the name of the course  
print(course.get_course_name())  
# Output: Introduction to Artificial Intelligence
```

2) Write a python program to show the Hierarchical inheritance of two or more classes named as “Square “ & “ Triangle” inherit from a single Base class as “Area “ .

```
class Area:  
    def __init__(self, length, width):  
        self.length = length  
        self.width = width
```

```
    def calculate_area(self):  
        return self.length * self.width
```

```
class Square(Area):
```

```
def __init__(self, length):  
    super().__init__(length, length)
```

```
class Triangle(Area):  
    def __init__(self, base, height):  
        super().__init__(base, height)
```

Override calculate_area method to calculate the area of a triangle

```
def calculate_area(self):  
    return 0.5 * self.length * self.width
```

Create a square with side length 4

```
square = Square(4)
```

```
print(square.calculate_area()) # Output: 16
```

Create a triangle with base 5 and height 6

```
triangle = Triangle(5, 6)
```

```
print(triangle.calculate_area()) # Output: 15
```

3) Define a class named Shape and its subclass (Square/Circle). The subclass has an init function which takes an argument (length/radius). Both classes have an area and volume function which can print the area and volume of the shape where Shape's area is 0 by default.

```
class Shape:
```

```
    def __init__(self):  
        self.area = 0
```

```
    def area(self):  
        return self.area
```

```
class Square(Shape):
    def __init__(self, length):
        self.length = length

    def area(self):
        return self.length * self.length

square = Square(5)
print(square.area()) # Output: 25
```

4) Python Program to Create a Class in which One Method Accepts a String from the User and Another method Prints it. Define a class named Country which has a method called print Nationality. Define subclass named state from Country which has a method called print State . Write a method to print state, country and nationality.

```
class Country:
    def __init__(self, nationality):
        self.nationality = nationality

    def print_nationality(self):
        print(f"The nationality is: {self.nationality}")

class State(Country):
    def __init__(self, state, nationality):
        super().__init__(nationality)
        self.state = state
```

```
def print_state(self):
    print(f"The state is: {self.state}")

def print_all(self):
    self.print_state()
    super().print_nationality()

state = State("New York", "American")
state.print_all()
```

SET C :

- 1) Write a Python Program to depict multiple inheritance when method is overridden in both classes and check the output accordingly.**

```
class A:
    def print_method(self):
        print("Method in class A")

class B:
    def print_method(self):
        print("Method in class B")

class C(A, B):
    def print_method(self):
        print("Method in class C")

c = C()
c.print_method() # Output: Method in class C
```

2) Write a Python Program to describe a HAS-A Relationship(Composition).

```
class Car:
```

```
    def __init__(self, make, model):  
        self.make = make  
        self.model = model
```

```
class Garage:
```

```
    def __init__(self, car):  
        self.car = car
```

```
car = Car("Ford", "Fiesta")  
garage = Garage(car)
```

```
print(garage.car.make)  
print(garage.car.model)
```

```
# Output: Ford
```

```
# Output: Fiesta
```


Assignment 7: Exception Handling

SET A :

- 1) Define a custom exception class which takes a string message as attribute.**

```
class CustomException(Exception):  
    def __init__(self, message):  
        self.message = message
```

```
raise CustomException("This is my custom error  
message")
```

```
class CustomException(Exception):  
    def __init__(self, message):  
        super().__init__(message)
```

```
raise CustomException("This is my custom error  
message")
```

- 2) Write a function called oops that explicitly raises a IndexError exception when called. Then write another function that calls oops inside a try/except statement to catch the error.**

```
def oops():  
    raise IndexError("An index error occurred")
```

```
def catch_error():  
    try:  
        oops()  
    except IndexError as error:  
        print("An error occurred:", error)
```

3) Change the oops function you just wrote to raise an exception you define yourself, called `MyError`, and pass an extra data item along with the exception. Then, extend the try statement in the catcher function to catch this exception and its data in addition to `IndexError`, and print the extra data item.

```
class MyError(Exception):
    def __init__(self, message, data):
        self.message = message
        self.data = data

def oops():
    raise MyError("An error occurred", {"extra_data": "foo"})

def catch_error():
    try:
        oops()
    except IndexError as error:
        print("An index error occurred:", error)
    except MyError as error:
        print("A custom error occurred:", error.message)
        print("Extra data:", error.data)
```

SET B :

1) Define a class Date(Day, Month, Year) with functions to accept and display it. Accept date from user. Throw user defined exception "invalidDateException" if the date is invalid.

class Date:

```
def __init__(self, day, month, year):  
    self.day = day  
    self.month = month  
    self.year = year
```

class Date:

```
def __init__(self, day, month, year):  
    self.day = day  
    self.month = month  
    self.year = year
```

```
def set_date(self):
```

```
    self.day = int(input("Enter the day: "))  
    self.month = int(input("Enter the month: "))  
    self.year = int(input("Enter the year: "))
```

class Date:

```
def __init__(self, day, month, year):  
    self.day = day  
    self.month = month  
    self.year = year
```

```
def set_date(self):
```

```
    self.day = int(input("Enter the day: "))  
    self.month = int(input("Enter the month: "))  
    self.year = int(input("Enter the year: "))
```

```
def display_date(self):  
    print("The date is:", self.day, self.month, self.year)
```

```
class invalidDateException(Exception):  
    def __init__(self, message):  
        self.message = message
```

```
class Date:  
    def __init__(self, day, month, year):  
        self.day = day  
        self.month = month  
        self.year = year  
  
    def set_date(self):  
        self.day = int(input("Enter the day: "))  
        self.month = int(input("Enter the month: "))  
        self.year = int(input("Enter the year: "))  
  
    # Check whether the entered date is valid  
    if self.day < 1 or self.day > 31:  
        raise invalidDateException("Invalid day")  
    if self.month < 1 or self.month > 12:  
        raise
```

2) Write text file named test.txt that contains integers, characters and float numbers. Write a Python program to read the test.txt file. And print appropriate message using exception.

try:

with open('test.txt') as f:

for line in f:

try:

Attempt to convert the line to an integer

value = int(line.strip())

print(f'{value} is an integer')

except ValueError:

try:

Attempt to convert the line to a float

value = float(line.strip())

print(f'{value} is a float')

except ValueError:

**# If the value is not an integer or a float, it must
be a character**

print(f'{line.strip()} is a character')

except FileNotFoundError:

print('File not found')

SET C:

1) Write a function called safe (func, *args) that runs any function using apply, catches any exception raised while the function runs, and prints the exception using the exc_type and exc_value attributes in the sys module. Then, use your safe function to run the oops function you wrote in Exercises 3. Put safe in a module file called tools.py, and pass it the oops function interactively. Finally, expand safe to also print a Python stack trace when an error occurs by calling the built-in print_exc() function in the standard traceback module (see the Python library reference manual or other Python books for details)

```
import sys
```

```
import traceback
```

```
def safe(func, *args):
```

```
    try:
```

```
        apply(func, args)
```

```
    except Exception as e:
```

```
        exc_type, exc_value = sys.exc_info()[:2]
```

```
        print(exc_type, exc_value)
```

```
        traceback.print_exc()
```

```
from tools import safe
```

```
def oops():
```

```
    x = 1/0
```

```
safe(oops)
```

2) Change the oops function in question 4 from SET A to raise an exception you define yourself, called `MyError`, and pass an extra data item along with the exception. You may identify your exception with either a string or a class. Then, extend the try statement in the catcher function to catch this exception and its data in addition to `IndexError`, and print the extra data item. Finally, if you used a string for your exception, go back and change it to be a class instance.

```
class MyError(Exception):
```

```
    pass
```

```
def oops():
```

```
    raise MyError("Something went wrong!")
```

```
import sys
```

```
import traceback
```

```
class MyError(Exception):
```

```
    pass
```

```
def safe(func, *args):
```

```
    try:
```

```
        apply(func, args)
```

```
    except (IndexError, MyError) as e:
```

```
        exc_type, exc_value = sys.exc_info()[:2]
```

```
        print(exc_type, exc_value)
```

```
        if isinstance(e, MyError):
```

```
            print("Extra data:", e.args[0])
```

```
        traceback.print_exc()
```

```
class MyError(Exception):
```

```
    def __init__(self, message, data):
```

```
        super().__init__(message)
```

```
self.data = data
```

```
def oops():  
    raise MyError("Something went wrong!", [1, 2, 3])
```

```
import sys  
import traceback
```

```
class MyError(Exception):  
    def __init__(self, message, data):  
        super().__init__(message)  
        self.data = data
```

```
def safe(func, *args):  
    try:  
        apply(func, args)  
    except (IndexError, MyError) as e:  
        exc_type, exc_value = sys.exc_info()[:2]  
        print(exc_type, exc_value)  
        if isinstance(e, MyError):  
            print("Extra data:", e.data)  
        traceback.print_exc()
```


Assignment 8: Python GUI Programming using Tkinter

Assignments:

Set A:

- 1. Write Python GUI program to display an alert message when a button is pressed.**

```
import tkinter as tk
```

```
# create the main window
```

```
root = tk.Tk()
```

```
# create a function that will be called when the button is pressed
```

```
def show_alert():
```

```
    # show an alert message
```

```
    tk.messagebox.showinfo("Alert", "Button pressed!")
```

```
# create a button and specify the function to call when it is pressed
```

```
button = tk.Button(root, text="Press me",  
command=show_alert)
```

```
# pack the button into the GUI window
```

```
button.pack()
```

```
# start the GUI event loop
```

```
root.mainloop()
```

2. Write Python GUI program to Create background with changing colors.

```
import tkinter as tk
```

```
# create the main window
```

```
root = tk.Tk()
```

```
# create a function that will be called repeatedly to change  
the background color
```

```
def change_color():
```

```
    # get the current color of the background
```

```
    current_color = root["bg"]
```

```
    # choose the next color based on the current color
```

```
    if current_color == "red":
```

```
        next_color = "green"
```

```
    elif current_color == "green":
```

```
        next_color = "blue"
```

```
    else:
```

```
        next_color = "red"
```

```
    # set the background color to the next color
```

```
    root["bg"] = next_color
```

```
    # schedule the next call to this function after a 1-second  
delay
```

```
    root.after(1000, change_color)
```

```
# schedule the first call to the change_color function
```

```
root.after(1000, change_color)
```

```
# start the GUI event loop
```

```
root.mainloop()
```

- 3. Write a Python GUI program to create a label and change the label font style (font name, bold, size) using tkinter module.**

```
import tkinter as tk
```

```
# create the main window
```

```
root = tk.Tk()
```

```
# create a label with some text
```

```
label = tk.Label(root, text="Hello, World!")
```

```
# set the font of the label to a custom font style
```

```
label["font"] = ("Arial", 16, "bold")
```

```
# pack the label into the GUI window
```

```
label.pack()
```

```
# start the GUI event loop
```

```
root.mainloop()
```

- 4. Write a Python GUI program to create a Text widget using tkinter module. Insert a string at the beginning then insert a string into the current text. Delete the first and last character of the text.**

```
import tkinter as tk
```

```
# create the main window
```

```
root = tk.Tk()
```

```
# create a text widget
```

```
text = tk.Text(root)
```

```
# insert some text at the beginning of the widget
```

```
text.insert("1.0", "Hello, World!\n")
```

```
# insert some more text at the current position in the widget
```

```
text.insert("insert", "This is some additional text.\n")
```

```
# delete the first and last character of the widget's text
```

```
text.delete("1.0", "1.1")
```

```
text.delete("end-1c", "end")
```

```
# pack the text widget into the GUI window
```

```
text.pack()
```

```
# start the GUI event loop
```

```
root.mainloop()
```

5. Write a Python GUI program to accept dimensions of a cylinder and display the surface area and volume of cylinder.

```
import tkinter as tk
```

```
import math
```

```
# create the main window
```

```
root = tk.Tk()
```

```
# create a function that will be called when the "Calculate" button is pressed
```

```
def calculate():
```

```
    # get the radius and height of the cylinder from the input fields
```

```
    radius = float(radius_input.get())
```

```
    height = float(height_input.get())
```

```
# calculate the surface area and volume of the cylinder
```

```
surface_area = 2 * math.pi * radius * (radius + height)
volume = math.pi * radius * radius * height

# display the results in the output fields
surface_area_output["text"] = f"Surface Area:
{surface_area:.2f}"
volume_output["text"] = f"Volume: {volume:.2f}"

# create the input fields for the radius and height of the
cylinder
radius_input = tk.Entry(root)
height_input = tk.Entry(root)

# create the "Calculate" button and specify the function to
call when it is pressed
calculate_button = tk.Button(root, text="Calculate",
command=calculate)

# create the output fields for the surface area and volume
of the cylinder
surface_area_output = tk.Label(root, text="Surface Area:")
volume_output = tk.Label(root, text="Volume:")

# pack the input fields, button, and output fields into the
GUI window
radius_input.pack()
height_input.pack()
calculate_button.pack()
surface_area_output.pack()
volume_output.pack()

# start the GUI event loop
root.mainloop()
```

6. Write Python GUI program that takes input string and change letter to upper case when a button is pressed.

import tkinter as tk

create the main window

root = tk.Tk()

create a function that will be called when the "Convert" button is pressed

def convert_to_uppercase():

get the input string from the input field

input_str = input_field.get()

convert the string to uppercase

output_str = input_str.upper()

display the converted string in the output field

output_field["text"] = output_str

create the input field for the input string

input_field = tk.Entry(root)

create the "Convert" button and specify the function to call when it is pressed

**convert_button = tk.Button(root, text="Convert",
command=convert_to_uppercase)**

create the output field for the converted string

output_field = tk.Label(root, text="")

pack the input field, button, and output field into the GUI window

input_field.pack()

```
convert_button.pack()  
output_field.pack()
```

```
# start the GUI event loop  
root.mainloop()
```

Set B:

1. Write Python GUI program to take input of your date of birth and output your age when a button is pressed.

```
import datetime
import tkinter as tk

# Create the main window
window = tk.Tk()
window.title("Age Calculator")

# Create the main frame
frame = tk.Frame(window)
frame.grid(column=0, row=0, sticky=(tk.N, tk.W, tk.E, tk.S))

# Create the entry field for the user to enter their date of birth
date_of_birth_label = tk.Label(frame, text="Enter your date of birth (YYYY-MM-DD):")
date_of_birth_label.grid(column=0, row=0, sticky=tk.W)
date_of_birth_entry = tk.Entry(frame)
date_of_birth_entry.grid(column=1, row=0, sticky=tk.W)

# Function to calculate the user's age when the "Calculate Age" button is pressed
def calculate_age():
    # Get the user's date of birth from the entry field
    date_of_birth_str = date_of_birth_entry.get()

    # Parse the date of birth string into a datetime object
    date_of_birth =
datetime.datetime.strptime(date_of_birth_str, "%Y-%m-%d")
```



```
# Calculate the user's age by subtracting their date of birth from the current date
```

```
age = datetime.datetime.now() - date_of_birth
```

```
# Output the user's age
```

```
print("Your age is:", age.days // 365)
```

```
# Create the "Calculate Age" button
```

```
calculate_age_button = tk.Button(frame, text="Calculate Age", command=calculate_age)
```

```
calculate_age_button.grid(column=0, row=1, columnspan=2, sticky=tk.W)
```

```
# Set the size of the window
```

```
window.columnconfigure(0, weight=1)
```

```
window.rowconfigure(0, weight=1)
```

```
# Start the main event loop
```

```
window.mainloop()
```

To run this program, save the code to a file named `age_calculator.py` and run the following command in a terminal or command prompt:

```
python age_calculator.py
```

2. Write Python GUI program which accepts a sentence from the user and alters it when a button is pressed. Every space should be replaced by *, case of all alphabets should be reversed, digits are replaced by ?.

```
import tkinter as tk
```

```
# Create the main window
```

```
window = tk.Tk()
```

```
window.title("Sentence Alterator")

# Create the main frame
frame = tk.Frame(window)
frame.grid(column=0, row=0, sticky=(tk.N, tk.W, tk.E, tk.S))

# Create the entry field for the user to enter their sentence
sentence_label = tk.Label(frame, text="Enter a sentence:")
sentence_label.grid(column=0, row=0, sticky=tk.W)
sentence_entry = tk.Entry(frame)
sentence_entry.grid(column=1, row=0, sticky=tk.W)

# Function to alter the user's sentence when the "Alter
Sentence" button is pressed
def alter_sentence():
    # Get the user's sentence from the entry field
    sentence = sentence_entry.get()

    # Replace spaces with *
    altered_sentence = sentence.replace(" ", "*")

    # Reverse the case of all alphabets
    altered_sentence = "".join([char.upper() if char.islower()
else char.lower() for char in altered_sentence])

    # Replace digits with ?
    altered_sentence = "".join([char if not char.isdigit() else "?"
for char in altered_sentence])

    # Output the altered sentence
    print("Altered sentence:", altered_sentence)

# Create the "Alter Sentence" button
```

```
alter_sentence_button = tk.Button(frame, text="Alter  
Sentence", command=alter_sentence)  
alter_sentence_button.grid(column=0, row=1, colspan=2,  
sticky=tk.W)
```

```
# Set the size of the window  
window.columnconfigure(0, weight=1)  
window.rowconfigure(0, weight=1)
```

```
# Start the main event loop  
window.mainloop()
```

To run this program, save the code to a file named `sentence_alterator.py` and run the following command in a terminal or command prompt:
`python sentence_alterator.py`

3. Write Python GUI A program to create a digital clock with Tkinter to display the time.

```
import tkinter as tk
```

```
# Create the main window  
window = tk.Tk()
```

```
# Create a label to display the time  
time_label = tk.Label(window)
```

```
# Set the time to be displayed as the text of the label  
time_label.configure(text="12:00:00")
```

```
# Add the label to the window  
time_label.pack()
```

```

# Function to update the time every second
def update_time():
    current_time = datetime.datetime.now()

    time_label.configure(text=current_time.strftime("%I:%M:%S %p"))
    window.after(1000, update_time)

# Start the clock
update_time()

# Start the main event loop
window.mainloop()

```

This code creates a digital clock with a Tkinter window that displays the current time in the format "hour:minute:second AM/PM". The time is updated every second by calling the `update_time()` function.

4. Create a program to generate a random password with upper and lower case letters.

```
import random
```

```

def generate_password():
    password = ""

    # Create a list of all uppercase and lowercase letters
    letters = [chr(i) for i in range(ord('A'), ord('Z')+1)] + [chr(i)
for i in range(ord('a'), ord('z')+1)]

    # Randomly choose 10 letters from the list and add them
to the password string

```

```
for i in range(10):  
    password += random.choice(letters)
```

```
return password
```

```
# Generate a password and print it  
password = generate_password()  
print(password)
```

5. Write Python GUI program which accepts a number n to displays each digit of number in words.

```
# Import the necessary modules  
import tkinter as tk
```

```
# Define a dictionary of digit names
```

```
digit_names = {  
    0: "zero",  
    1: "one",  
    2: "two",  
    3: "three",  
    4: "four",  
    5: "five",  
    6: "six",  
    7: "seven",  
    8: "eight",  
    9: "nine"  
}
```

```
# Define a function to convert a number to words
```

```
def number_to_words(n):
```

```
    # Convert the number to a string of digits  
    digits = str(n)
```

```
    # Initialize an empty list to store the digit names
```

```
digit_words = []

# Loop through each digit in the number
for digit in digits:
    # Convert the digit to an integer and look up the
    corresponding word
    digit_word = digit_names[int(digit)]

    # Add the word to the list of digit words
    digit_words.append(digit_word)

# Join the words together with spaces and return the
result
return " ".join(digit_words)

# Create a Tkinter window
window = tk.Tk()

# Set the window title
window.title("Number to Words")

# Create a label to prompt the user to enter a number
label = tk.Label(window, text="Enter a number:")
label.pack(side=tk.LEFT)

# Create an entry field to accept the number
entry = tk.Entry(window)
entry.pack(side=tk.LEFT)

# Create a button to convert the number to words
button = tk.Button(window, text="Convert")
button.pack(side=tk.LEFT)
```

```

# Define a function to handle button clicks
def on_button_click():
    # Get the number from the entry field
    number = entry.get()

    # Convert the number to words
    words = number_to_words(number)

    # Display the words in a message box
    tk.messagebox.showinfo("Number to Words", words)

# Tell the button to call the on_button_click function
when clicked
button.configure(command=on_button_click)

# Start the event loop
window.mainloop()

```

6. Write Python GUI program to accept a decimal number and convert and display it to binary, octal and hexadecimal number.

```

# Import the necessary modules
import tkinter as tk

# Create a Tkinter window
window = tk.Tk()

# Set the window title
window.title("Number Converter")

# Create a label to prompt the user to enter a decimal
number
label = tk.Label(window, text="Enter a decimal
number:")

```

```
label.pack(side=tk.LEFT)
```

```
# Create an entry field to accept the decimal number
```

```
entry = tk.Entry(window)
```

```
entry.pack(side=tk.LEFT)
```

```
# Create a button to convert the number
```

```
button = tk.Button(window, text="Convert")
```

```
button.pack(side=tk.LEFT)
```

```
# Define a function to handle button clicks
```

```
def on_button_click():
```

```
    # Get the decimal number from the entry field
```

```
    decimal = int(entry.get())
```

```
    # Convert the decimal number to binary, octal, and  
    hexadecimal
```

```
    binary = bin(decimal)
```

```
    octal = oct(decimal)
```

```
    hexadecimal = hex(decimal)
```

```
    # Display the results in a message box
```

```
    tk.messagebox.showinfo("Number Converter",  
f"Binary: {binary}\nOctal: {octal}\nHexadecimal:  
{hexadecimal}")
```

```
# Tell the button to call the on_button_click function  
when clicked
```

```
button.configure(command=on_button_click)
```

```
# Start the event loop
```

```
window.mainloop()
```


7. Write Python GUI program to add items in listbox widget to print and delete the selected items from listbox on button click. Provide two separate button for print and delete.

```
# Import the necessary modules  
import tkinter as tk
```

```
# Create a Tkinter window  
window = tk.Tk()
```

```
# Set the window title  
window.title("List Box Demo")
```

```
# Create a list box widget  
listbox = tk.Listbox(window)  
listbox.pack(side=tk.LEFT)
```

```
# Create a button to add items to the list box  
add_button = tk.Button(window, text="Add")  
add_button.pack(side=tk.LEFT)
```

```
# Define a function to handle the Add button click  
def on_add_click():  
    # Get the text from the entry field  
    text = entry.get()
```

```
# Insert the text into the list box at the end  
listbox.insert(tk.END, text)
```

```
# Tell the Add button to call the on_add_click function  
when clicked  
add_button.configure(command=on_add_click)
```

Create an entry field to accept the text to add to the list box

```
entry = tk.Entry(window)  
entry.pack(side=tk.LEFT)
```

Create a button to print the selected item

```
print_button = tk.Button(window, text="Print")  
print_button.pack(side=tk.LEFT)
```

Define a function to handle the Print button click

```
def on_print_click():
```

Get the index of the selected item in the list box

```
index = listbox.curselection()
```

Check if an item is selected

if index:

Get the text of the selected item

```
text = listbox.get(index)
```

Print the text in a message box

```
tk.messagebox.showinfo("Selected Item", text)
```

else:

If no item is selected, display an error message

```
tk.messagebox.showerror("Error", "No item selected.")
```

Tell the Print button to call the on_print_click function when clicked

```
print_button.configure(command=on_print_click)
```

Create a button to delete the selected item

```
delete_button = tk.Button(window, text="Delete")
```

```
delete_button.pack(side=tk.LEFT)
```

```

# Define a function to handle the Delete button click
def on_delete_click():
    # Get the index of the selected item in the list box
    index = listbox.curselection()

    # Check if an item is selected
    if index:
        # Delete the selected item from the list box
        listbox.delete(index)
    else:
        # If no item is selected, display an error message
        tk.messagebox.showerror("Error", "No item selected.")

# Tell the Delete button to call the on_delete_click function
when clicked
delete_button.configure(command=on_delete_click)

# Start the event loop
window.mainloop()

```

8. Write Python GUI program to add menu bar with name of colors as options to change the background color as per selection from menu option.

```

# Import the necessary modules
import tkinter as tk

# Create a Tkinter window
window = tk.Tk()

# Set the window title
window.title("Color Changer")

# Create a menu bar
menubar = tk.Menu(window)

```

```
# Create a menu named "Colors"
colors_menu = tk.Menu(menubar, tearoff=0)
menubar.add_cascade(label="Colors", menu=colors_menu)
```

```
# Define a function to change the background color
def change_color(color):
    # Set the background color of the window
    window.configure(background=color)
```

```
# Add color options to the Colors menu
colors_menu.add_command(label="Red",
    command=lambda: change_color("red"))
colors_menu.add_command(label="Green",
    command=lambda: change_color("green"))
colors_menu.add_command(label="Blue",
    command=lambda: change_color("blue"))
colors_menu.add_command(label="Yellow",
    command=lambda: change_color("yellow"))
```

```
# Attach the menu bar to the window
window.config(menu=menubar)
```

```
# Start the event loop
window.mainloop()
```

9. Write Python GUI program to accept a number n and check whether it is Prime, Perfect or Armstrong number or not. Specify three radio buttons.

```
# Import the necessary modules
import tkinter as tk
```

```
# Create a Tkinter window
window = tk.Tk()
```

```
# Set the window title
window.title("Number Checker")

# Create a label to prompt the user to enter a number
label = tk.Label(window, text="Enter a number:")
label.pack(side=tk.LEFT)

# Create an entry field to accept the number
entry = tk.Entry(window)
entry.pack(side=tk.LEFT)

# Create a variable to store the selected radio button
selected = tk.StringVar()

# Create a radio button for checking if a number is prime
prime_radio = tk.Radiobutton(window, text="Prime",
variable=selected, value="prime")
prime_radio.pack(side=tk.LEFT)

# Create a radio button for checking if a number is perfect
perfect_radio = tk.Radiobutton(window, text="Perfect",
variable=selected, value="perfect")
perfect_radio.pack(side=tk.LEFT)

# Create a radio button for checking if a number is an
Armstrong number
armstrong_radio = tk.Radiobutton(window,
text="Armstrong", variable=selected, value="armstrong")
armstrong_radio.pack(side=tk.LEFT)

# Create a button to check the number
check_button = tk.Button(window, text="Check")
```

```
check_button.pack(side=tk.LEFT)
```

```
# Define a function to handle the Check button click
```

```
def on_check_click():
```

```
    # Get the number from the entry field
```

```
    number = int(entry.get())
```

```
    # Check which radio button is selected
```

```
    if selected.get() == "prime":
```

```
        # Check if the number is prime
```

```
        is_prime = check_if_prime(number)
```

```
        result = "prime" if is_prime else "not prime"
```

```
    elif selected.get() == "perfect":
```

```
        # Check if the number is perfect
```

```
        is_perfect = check_if_perfect(number)
```

```
        result = "perfect" if is_perfect else "not perfect"
```

```
    elif selected.get() == "armstrong":
```

```
        # Check if the number is an Armstrong number
```

```
        is_armstrong = check_if_armstrong(number)
```

```
        result = "an Armstrong number" if is_armstrong else "not  
an Armstrong number"
```

```
    # Display the result in a message box
```

```
    tk.messagebox.showinfo("Number Checker", f"The  
number {number} is {result}.")
```

```
# Tell the Check button to call the on_check_click function  
when clicked
```

```
check_button.configure(command=on_check_click)
```

```
# Define a function to check if a number is prime
```

```
def check_if_prime(n):
```

```
    if n <= 1:
```

```
    return False
for i in range(2, n):
    if n % i == 0:
        return False
return True
```

Define a function to check if a number is perfect

```
def check_if_perfect(n):
```

10. Write a Python GUI program to create a label and change the label font style (font name, bold, size). Specify separate checkbutton for each style.

Import the necessary modules

```
import tkinter as tk
```

Create a Tkinter window

```
window = tk.Tk()
```

Set the window title

```
window.title("Font Style Demo")
```

Create a label

```
label = tk.Label(window, text="Hello, world!")
```

```
label.pack()
```

Create a variable to store the font name

```
font_name = tk.StringVar()
```

Create a check button for changing the font name

```
font_check = tk.Checkbutton(window, text="Font",
variable=font_name)
```

```
font_check.pack()
```

Create a variable to store the font bold setting

```
font_bold = tk.IntVar()
```

```
# Create a check button for toggling font boldness  
bold_check = tk.Checkbutton(window, text="Bold",  
variable=font_bold)  
bold_check.pack()
```

```
# Create a variable to store the font size  
font_size = tk.IntVar()
```

```
# Create a check button for changing the font size  
size_check = tk.Checkbutton(window, text="Size",  
variable=font_size)  
size_check.pack()
```

```
# Define a function to handle check button clicks  
def on_check():  
    # Get the font name, boldness, and size from the check  
    buttons  
    name = font_name.get()  
    bold = font_bold.get()  
    size = font_size.get()
```

```
# Compute the font string  
font = (name, size, "bold") if bold else (name, size)
```

```
# Set the font of the label  
label.configure(font=font)
```

```
# Tell the check buttons to call the on_check function when  
clicked  
font_check.configure(command=on_check)  
bold_check.configure(command=on_check)
```



```
size_check.configure(command=on_check)
```

```
# Start the event loop  
window.mainloop()
```

Set C:

1. Write a Python GUI program to implement simple calculator.

```
import tkinter as tk
```

```
class Calculator(tk.Frame):
```

```
    def __init__(self, parent):
```

```
        tk.Frame.__init__(self, parent)
```

```
        self.create_widgets()
```

```
    def create_widgets(self):
```

```
        self.input_field = tk.Entry(self, width=40,  
font=('Helvetica', 24))
```

```
        self.input_field.pack(side=tk.TOP, padx=10, pady=10)
```

```
        button_frame = tk.Frame(self)
```

```
        button_frame.pack(side=tk.TOP, padx=10, pady=10)
```

```
    # Create a row of buttons for each row of the keyboard
```

```
    for i in range(4):
```

```
        row = []
```

```
        for j in range(3):
```

```

        button = tk.Button(button_frame, text=str(3 * i + j
+ 1),
                            font=('Helvetica', 24),
                            command=self.button_press)
        button.grid(row=i, column=j, sticky='nesw')
        row.append(button)
        self.buttons.append(row)

```

Create the final row of buttons

```

row = []
for j in range(2):
    button = tk.Button(button_frame, text=str(3 * 4 + j +
1),
                        font=('Helvetica', 24),
                        command=self.button_press)
    button.grid(row=4, column=j, sticky='nesw',
columnspan=2)
    row.append(button)
    self.buttons.append(row)

```

Create the special buttons

```

self.buttons.append([tk.Button(button_frame, text='0',
font=('Helvetica', 24),
                            command=self.button_press)])
self.buttons[-1][0].grid(row=5, column=0,
sticky='nesw')

```

```

self.buttons.append([tk.Button(button_frame, text='.',
font=('Helvetica', 24),
                            command=self.button_press)])
self.buttons[-1][0].grid(row=5, column=1,
sticky='nesw')

```

```
self.buttons.append([tk.Button(button_frame, text='+',
font=('Helvetica', 24),
                        command=self.button_press)])
self.buttons[-1][0].grid(row=5, column=2,
sticky='nesw')
```

```
def button_press(self):
    # Get the text of the button that was pressed
    button_text = self.sender.cget('text')

    # If the button text is a digit, append it to the input
    field
    if button_text.isdigit():
        self.input_field.insert(tk.END, button_text)

    # If the button text is '.', only append it if there's not
    already
    # a decimal point in the input field
    elif button_text == '.':
        input_text = self.input_field.get()
```

Here's a simple Python program that uses the tkinter module to create a basic calculator GUI.