

CSA0980 -- PROGRAMMING IN JAVA FOR IDL TECHNOLOGY :-

M . NAGA KESHAHA

REG NO :-192125019

1. Develop an Event Handling Applet Program in Java to print a message When the button is clicked.

PROGRAM :-

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class EventHandlingApplet extends Applet implements ActionListener {
    private Button button;

    public void init() {
        button = new Button("Click me!");
        add(button);
        button.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button) {
            System.out.println("Button clicked!");
        }
    }
}
```

2. Generate a Java Code to Write and Read the String "WELCOME TO SSE" using FileOutputStream and FileInputStream class.

Debugging

PROGRAM :-

```
import java.util.*;
public class FileReadWriteExample
{
    public static void main(String[] args)
    {
        String message = "WELCOME TO SSE";
        String fileName = "file.txt";
        try
```

```

{
    FileOutputStream fileOutputStream = new FileOutputStream(fileName);
    fileOutputStream.write(message.getBytes());
    fileOutputStream.close();
    System.out.println("Message written to file.");
} catch (IOException e) {
    System.out.println("An error occurred while writing to file: " +
e.getMessage());
    return;
}
try
{
    FileInputStream fileInputStream = new FileInputStream(fileName);
    byte[] buffer = new byte[1024];
    int length = fileInputStream.read(buffer);
    String messageFromFile = new String(buffer, 0, length);
    fileInputStream.close();
    System.out.println("Message read from file: " + messageFromFile);
} catch (IOException e) {
    System.out.println("An error occurred while reading from file: " +
e.getMessage());
}
}
}
}

```

3. We define the usage of capitals in a word to be right when one of the following cases holds:

All letters in this word are capitals, like "USA".

All letters in this word are not capitals, like "leetcode".

Only the first letter in this word is capital, like "Google".

Given a string word, return true if the usage of capitals in it is right.

Example 1:

Input: word = "USA"

Output: true

Example 2:

Input: word = "FlaG"

Output: false

Constraints:

1 ≤ word.length ≤ 100

word consists of lowercase and uppercase English letters.

```

class Solution {
    bool detectCapitalUse(string word) {

    }
}

```

PROGRAM :-

```
class Solution {
    public boolean detectCapitalUse(String word) {
        return word.matches("[A-Z]+|[a-z]+|[A-Z][a-z]+");
    }
}
```

4. You are given an array of characters letters that is sorted in non-decreasing order, and a character target. There are at least two different characters in letters. Return the smallest character in letters that is lexicographically greater than target. If such a character does not exist, return the first character in letters.

Example 1:

Input: letters = ["c","f","j"], target = "a";

Output: "c";

Explanation: The smallest character that is lexicographically greater than 'a' in letters is 'c'.

Example 2:

Input: letters = ["c","f","j"], target = "c";

Output: "f";

Explanation: The smallest character that is lexicographically greater than 'c' in letters is 'f'.

Example 3:

Input: letters = ["x","x","y","y"], target = "z";

Output: "x";

Explanation: There are no characters in letters that is lexicographically greater than 'z', so we return letters[0].

Constraints:

2 ≤ letters.length ≤ 104

letters[i] is a lowercase English letter.

letters is sorted in non-decreasing order.

letters contains at least two different characters.

target is a lowercase English letter.

```
class Solution {
    char nextGreatestLetter(char[] letters, char target) {

    }
}
```

PROGRAM :-

```
class Solution {
    public char nextGreatestLetter(char[] letters, char target) {
        int left = 0;
        int right = letters.length - 1;
```

```

char result = letters[0]; // set the default result to the first character

while (left <= right) {
    int mid = left + (right - left) / 2;
    if (letters[mid] > target) {
        result = letters[mid];
        right = mid - 1;
    } else {
        left = mid + 1;
    }
}
return result;
}
}

```

5. Program to show syntax of conditional and looping statement by menu choice : Find/Debug error in following code

```

import java.util.Scanner;
public class Menusel
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        char choice;
        do
        {
            System.out.println(Help on : &quot;);
            System.out.println(&quot;1. if&quot;);
            System.out.println(&quot;2. switch&quot;);
            System.out.println(&quot;3. while&quot;);
            System.out.println(&quot;4. do-while&quot;);
            System.out.println(&quot;5. for\n&quot;);
            System.out.println(&quot;Choose any one : &quot;);
            choice = scan.next().charAt(0);
        }while(choice <= '1' & & choice >= '5');
        System.out.println(&quot;\n&quot;);
        switch(choice)
        {
            case '1' : System.out.println(&quot;The if :\n&quot;);
                System.out.println(&quot;if(condition)\n{\ntstatement\n}&quot;);
                System.out.println(&quot;else\n{\n\ntstatement\n}&quot;);
                break;
            case '2' : System.out.println(&quot;The switch :\n&quot;);
                System.out.println(&quot;switch(expression)\n{&quot;);
                System.out.println(&quot;\tcase constant: statement sequence\n\tbreak;&quot;);
                System.out.println(&quot;\t//...\n&quot;);

```

```

break;
case &#39;3&#39; : System.out.println(&quot;The while :\n&quot;);
System.out.println(&quot;while(condition)\n{&quot;);
System.out.println(&quot;\t// body of loop\n}&quot;);
break;
case &#39;4&#39; : System.out.println(&quot;The do-while :\n&quot;);
System.out.println(&quot;do\n{&quot;);
System.out.println(&quot;\t// body of loop\n\n}while(condition);&quot;);

case &#39;5 : System.out.println(&quot;The for :\n&quot;);
System.out.println(&quot;for(initialization; condition; iteration)\n{&quot;);
System.out.println(&quot;\t// body of loop\n}&quot;);
break;
}
}
}

```

PROGRAM :-

```

import java.util.Scanner;
public class Menusel
{
    public static void main(String args[])
    {
        Scanner scan = new Scanner(System.in);
        charchoice;
        do
        {
            System.out.println("Help on:");
            System.out.println("1. if");
            System.out.println("2. switch");
            System.out.println("3. while");
            System.out.println("4. do-while");
            System.out.println("5. for\n");

            System.out.println("Choose any one: ");
            choice = scan.next().charAt(0);
        } while(choice < '1' || choice > '5');

        System.out.println("\n");

        switch(choice) {
            case '1':
                System.out.println("The if:\n");
                System.out.println("if(condition)\n{\n\tstatement\n}");
                System.out.println("else\n{\n\tstatement\n}");
                break;

```

```

case '2':
System.out.println("The switch:\n");
System.out.println("switch(expression)\n{");
System.out.println("\tcase constant: statement sequence\n\tbreak;");
System.out.println("\t//...\n}");
break;

case '3':
System.out.println("The while:\n");
System.out.println("while(condition)\n{");
System.out.println("\t// body of loop\n}");
break;

case '4':
System.out.println("The do-while:\n");
System.out.println("do\n{");
System.out.println("\t// body of loop\n\n} while(condition);");
break;

case '5':
System.out.println("The for:\n");
System.out.println("for(initialization; condition; iteration)\n{");
System.out.println("\t// body of loop\n}");
break;
}
}
}

```

6. Write a program to count all the prime and composite numbers entered by the user.

Sample Input:

Enter the numbers

4
54
29
71
7
59
98
23

Sample Output:

Composite number:3

Prime number:5

Test cases:

1. 33, 41, 52, 61,73,90

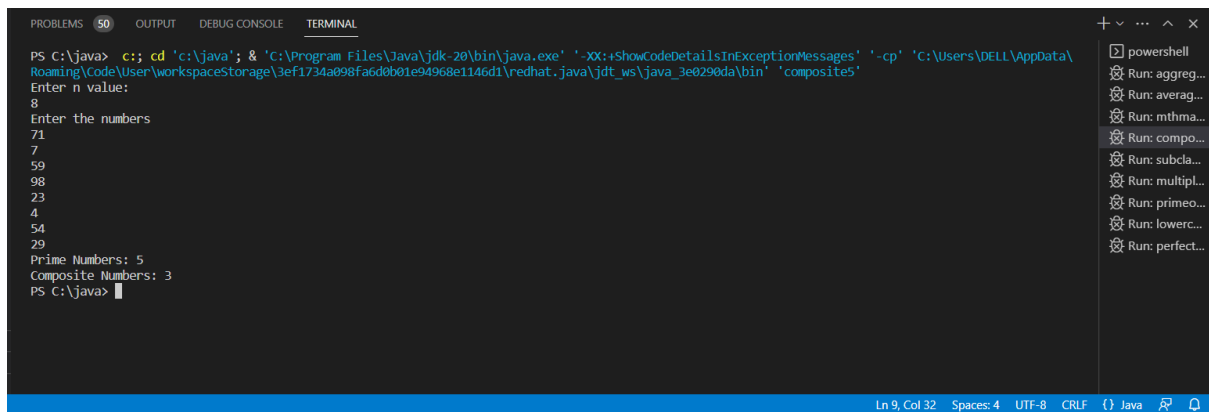
2. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE

3. 45, 87, 09, 5.0 ,2.3, 0.4
4. -54, -76, -97, -23, -33, -98
5. 45, 73, 00, 50, 67, 44

PROGRAM :-

```
import java.util.*;
public class composite5 {
    public static void main(String[] args) {
        Scanner z = new Scanner(System.in);
        System.out.println("Enter n value:");
        int n = z.nextInt();
        int a[] = new int[n];
        System.out.println("Enter the numbers");
        for(int i=0;i<n;i++)
        {
            a[i] = z.nextInt();
        }
        int p=0,c=0;
        for(int i=0;i<n;i++)
        {
            if(isPrime(a[i]))
            {
                p++;
            }
            else
            {
                c++;
            }
        }
        System.out.println("Prime Numbers: "+p);
        System.out.println("Composite Numbers: "+c);
    }
    public static boolean isPrime(int n)
    {
        if(n<=1)
        {
            return false;
        }
        for(int i=2;i<=Math.sqrt(n);i++)
        {
            if(n%i==0)
            return false;
        }
        return true;
    }
}
```

OUTPUT :-



```
PS C:\java> cd 'c:\java'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e029eda\bin' 'composites'
Enter n value:
8
Enter the numbers
71
7
59
98
23
4
54
29
Prime Numbers: 5
Composite Numbers: 3
PS C:\java>
```

7. Find the M th maximum number and N th minimum number in an array and then find the sum of it and difference of it.

Sample Input:

Array of elements = {14, 16, 87, 36, 25, 89, 34}

M = 1

N = 3

Sample Output:

1 st Maximum Number = 89

3 rd Minimum Number = 25

Sum = 114

Difference = 64

Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1
2. {0, 0, 0, 0}, M = 1, N = 2
3. {-12, -78, -35, -42, -85}, M = 3 , N = 3
4. {15, 19, 34, 56, 12}, M = 6 , N = 3
5. {85, 45, 65, 75, 95}, M = 5 , N = 7

PROGRAM :-

```
import java.util.*;
public class mthmaximum5 {

    public static void main(String[] args) {
        int[] arr = {14, 16, 87, 36, 25, 89, 34};
        int m = 1;
        int n = 3;
        Arrays.sort(arr);
        int mthMax = arr[arr.length - m];
        int nthMin = arr[n - 1];
        int sum = mthMax + nthMin;
        int diff = mthMax - nthMin;
        System.out.println("1st Maximum Number = " + mthMax);
        System.out.println(n + "th Minimum Number = " + nthMin);
    }
}
```



```

        System.out.println("Sum = " + sum);
        System.out.println("Difference = " + diff);
    }
}

```

OUTPUT :-

```

PS C:\java> cd 'c:\java'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'mthmaximum5'
1st Maximum Number = 89
3th Minimum Number = 25
Sum = 114
Difference = 64
PS C:\java>

```

8. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than \$10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

Sample Input & Output:

Enter the grade of the employee: B

Enter the employee salary: 50000

Salary=50000

Bonus=5000.0

Total to be paid:55000.0

Test cases:

1. Enter the grade of the employee: A

Enter the employee salary: 8000

2. Enter the grade of the employee: C

Enter the employee salary: 60000

3. Enter the grade of the employee: B

Enter the employee salary: 0

4. Enter the grade of the employee: 38000

Enter the employee salary: A

5. Enter the grade of the employee: B

Enter the employee salary: -8000

PROGRAM :-

```

import java.util.Scanner;
public class bonus5 {

```

```

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter the grade of the employee: ");
    String grade = input.nextLine();
    System.out.print("Enter the employee salary: ");
    double salary = input.nextDouble();
    double bonus = 0;
    if (grade.equalsIgnoreCase("A")) {
        bonus = 0.05 * salary;
    } else if (grade.equalsIgnoreCase("B")) {
        bonus = 0.1 * salary;
    } else {
        System.out.println("Invalid grade entered.");
        System.exit(0);
    }
    if (salary < 10000) {
        bonus += 0.02 * salary;
    }
    double totalSalary = salary + bonus;
    System.out.printf("Salary=%.1f\n", salary);
    System.out.printf("Bonus=%.1f\n", bonus);
    System.out.printf("Total to be paid=%.1f\n", totalSalary);
}
}

```

OUTPUT :-

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\java> & 'C:\Program Files\Java\jdk-20\bin\java.exe' ^-XX:+ShowCodeDetailsInExceptionMessages' ^-cp 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098faed0b01e94968e1146d1\redhat.java\jdk_ws\java_3e0290da\bin' ^'bonus5'
Enter the grade of the employee: B
Enter the employee salary: 50000
Salary=50000.0
Bonus=5000.0
Total to be paid=55000.0
PS C:\java>

```

9. Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors)

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0

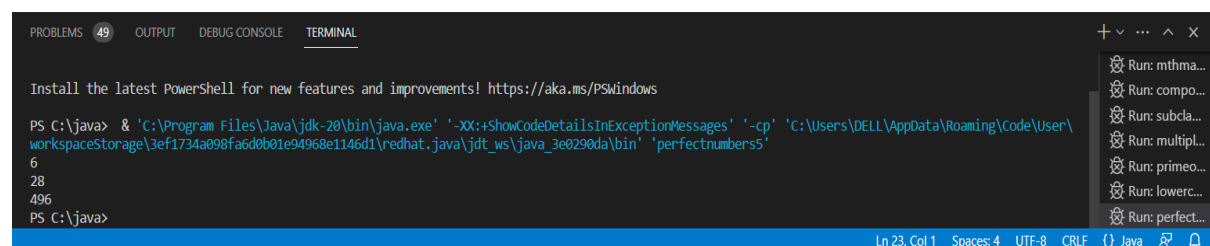
2. N = 5

3. N = -2
4. N = -5
5. N = 0.2

PROGRAM :-

```
public class perfectnumbers5 {
    public static void main(String[] args) {
        int n = 3;
        int count = 0;
        int num = 1;
        while (count < n) {
            int sum = 0;
            for (int i = 1; i < num; i++)
            {
                if (num % i == 0) {
                    sum += i;
                }
            }
            if (sum == num)
            {
                System.out.println(num);
                count++;
            }
            num++;
        }
    }
}
```

OUTPUT :-



```
PS C:\java> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'perfectnumbers5'
6
28
496
PS C:\java>
```

10. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60 >= and < 75, then the grade is First Division. If aggregate is 50 >= and < 60, then the grade is Second Division. If aggregate is 40 >= and < 50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

Test cases:

a) 18, 76,93,65

b) 73,78,79,75

c) 98,106,120,95

d) 96,73, -85,95

e) 78,59.8,76,79

PROGRAM :-

```
import java.util.*;
public class aggregate5 {
    public static void main(String[] args) {
        int a,b,c,d,total;
        Scanner v = new Scanner (System.in);
        System.out.println("Enter the Python : ");
        a = v.nextInt();
        System.out.println("Enter the C programming : ");
        b = v.nextInt();
        System.out.println("Enter the Mathematics : ");
        c = v.nextInt();
        System.out.println("Enter the Physics : ");
        d = v.nextInt();
        total = a+b+c+d;
        System.out.println("T0tal Subjects marks : " + total);
        double cs = total/4.0;
        System.out.println("Aggregate : "+ cs);
        if(cs >= 75)
        {
            System.out.println("DISTINCTION");
        }
        else if(cs >= 60 && cs <= 75)
        {
            System.out.println("First Division");
        }
        else if(cs >= 50 && cs <=60)
        {
            System.out.println("Second Division ");
        }
        else if(cs >= 40 && cs <= 50)
        {
            System.out.println("third Division");
        }
    }
}
```

```

    }
    else
    {
        System.out.println("Fail");
    }

}

}

```

OUTPUT :-

```

Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'aggregate5'
Enter the Python :
90
Enter the C programming :
91
Enter the Mathematics :
92
Enter the Physics :
93
Total Subjects marks : 366
Aggregate : 91.5
DISTINCTION
PS C:\java>

```

11. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.

Sample Input:

Enter -1 to exit...

Enter the number: 7

Enter the number: -2

Enter the number: 9

Enter the number: -8

Enter the number: -6

Enter the number: -4

Enter the number: 10

Enter the number: -1

Sample Output:

The average of negative numbers is: -5.0

The average of positive numbers is : 8.666666667

Test cases:

1. -1,43, -87, -29, 1, -9

2. 73, 7-6,2,10,28,-1

3. -5, -9, -46,2,5,0

4. 9, 11, -5, 6, 0, -1

5. -1, -1, -1, -1, -1

PROGRAM :-

```
import java.util.*;
public class averagepositive5 {
    public static void main(String[] args) {
        Scanner z = new Scanner(System.in);
        int psum = 0;
        int pcount = 0;
        int nsum = 0;
        int ncount = 0;
        while(true)
        {
            System.out.println("Enter a number(-1 to exist)");
            int n = z.nextInt();
            if(n== -1)
            {
                break;
            }
            else if(n>0)
            {
                psum+=n;
                pcount++;
            }
            else
            {
                nsum+=n;
                ncount++;
            }
        }
        if(pcount>0)
        {
            double pavg= (double)psum/pcount;
            System.out.println("Average of positive numbers:"+pavg);
        }
        else
        {
            System.out.println("No positive number is entered");
        }
        if(ncount>0)
        {
            double navg =(double)nsum/ncount;
            System.out.println("Average of negative numbers:"+navg);
        }
        else
        {
            System.out.println("No negative number is entered");
        }
    }
}
```

```

        System.out.println("No negative numbers entered:");
    }
    z.close();
}
}

```

OUTPUT :-

```

PS C:\java> cd 'c:\java'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'averagepositive5'
Enter a number(-1 to exist)
6
Enter a number(-1 to exist)
9
Enter a number(-1 to exist)
3
Enter a number(-1 to exist)
5
Enter a number(-1 to exist)
-5
Enter a number(-1 to exist)
-6
Enter a number(-1 to exist)
-1
Average of positive numbers:5.75
Average of negative numbers:-5.5
PS C:\java>

```

12. Write a program to read a character until a * is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.

Sample Input:

Enter * to exit...

Enter any character: W

Enter any character: d

Enter any character: A

Enter any character: G

Enter any character: g

Enter any character: H

Enter any character: *

Sample Output:

Total count of lower case:2

Total count of upper case:4

Total count of numbers =0

Test cases:

1. 1,7,6,9,5

2. S, Q, I, K,7, j, M

3. M, j, L, &, @, G

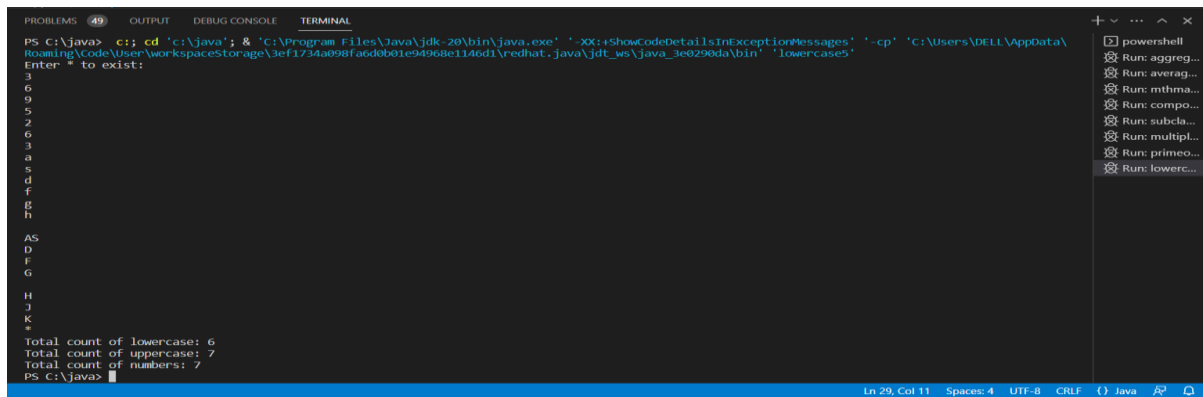
4. D, K, I, 6, L, *

5. *, K, A, e, 1, 8, %, *

PROGRAM :-

```
import java.util.*;
public class lowercase5 {
    public static void main(String[] args)
    {
        Scanner z = new Scanner(System.in);
        int ucount=0;
        int lcount=0;
        int num=0;
        int sp=0;
        System.out.println("Enter * to exist: ");
        while(true)
        {
            char c = z.next().charAt(0);
            if(c=='*')
            {
                break;
            }
            if(Character.isUpperCase(c))
            {
                ucount++;
            }
            else if(Character.isLowerCase(c))
            {
                lcount++;
            }
            else if(Character.isDigit(c))
            {
                num++;
            }
            else
            {
                sp++;
            }
        }
        System.out.println("Total count of lowercase: "+lcount);
        System.out.println("Total count of uppercase: "+ucount);
        System.out.println("Total count of numbers: "+num);
    }
}
```

OUTPUT :-



```
PS C:\java> cd 'c:\java'; & 'c:\program files\java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1754a058fadd8b01e94968e1146d1\redhat-java\jdk_ws\java_3e0290da\bin' 'lowercase'
Enter * to exist:
3
6
9
5
2
6
3
a
s
d
f
g
h
AS
D
F
G
H
J
K
*
Total count of lowercase: 6
Total count of uppercase: 7
Total count of numbers: 7
PS C:\java>
```

13. Using the concepts of thread with implementing Runnable interface in Java to find whether a given number is prime or not.

Sample Input : 5

Sample Output : 5 is Prime

Sample Output : 15

Test Cases

1. 4
2. -10
3. 0
4. EIGHT SEVEN
5. 11.48

PROGRAM :-

```
class CheckPrime implements Runnable {
    private int number;

    public CheckPrime(int number) {
        this.number = number;
    }

    public void run() {
        boolean isPrime = true;
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            System.out.println(number + " is prime");
        } else {
            System.out.println(number + " is not prime");
        }
    }
}
```

```

public class primeornot5 {
    public static void main(String[] args) {
        int number = 15;
        Thread thread = new Thread(new CheckPrime(number));
        thread.start();
    }
}

```

OUTPUT :-

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\java> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'primeornot5'
15 is not prime
PS C:\java>

```

14. Bring out the situation in which member names of a subclass hide members by the same name in the super class. How it can be resolved? Write Suitable code in Java and Implement above scenario with the Parametrized Constructor (accept int type parameter) of the Super Class can be called from Sub Class Using super () and display the input values provided.

Sample Input : 100, 200

Sample Output : 100, 200

Test Cases

1. 10, 20
2. -20, -30
3. 0, 0
4. EIGHT FIVE
5. 10.57, 12.58

PROGRAM :-

```

class Superclass {
    int x;
    public Superclass(int x) {
        this.x = x;
    }
}
class Subclass extends Superclass {
    int y;
    public Subclass(int x, int y) {
        super(x);
        this.y = y;
    }
}

```

```

    }
    void printValues() {
        System.out.println("x: " + super.x);
        System.out.println("y: " + y);
    }
}
public class subclasses5 {
    public static void main(String[] args) {
        Subclass s = new Subclass(100, 200);
        s.printValues();
    }
}

```

OUTPUT :-

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\java> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098fa6d0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'subclasses5'
x: 100
y: 200
PS C:\java>

```

15. Display Multiplication table for 5 and 10 using various stages of life cycle of the thread by generating a suitable code in Java.

Sample Input 5, 10

5 X 1 = 5

5 X 2 =10

....

10 X 1 =10

10 X 2 = 20

....

Test Cases:

1. 10, 20

2. -10, -30

3. 0, 0

4. SIX, SIX

9.8, 9.6

PROGRAM :-

```

class MultiplicationTable implements Runnable {
    private int number;

    public MultiplicationTable(int number) {
        this.number = number;
    }
}

```

```

    }
    public void run() {
        System.out.println("Multiplication table for " + number);
        for (int i = 1; i <= 10; i++) {
            System.out.println(number + " X " + i + " = " + (number * i));
        }
    }
}

public class multiplication5 {
    public static void main(String[] args) throws InterruptedException {
        Thread thread1 = new Thread(new MultiplicationTable(5));
        Thread thread2 = new Thread(new MultiplicationTable(10));
        thread1.start();
        thread2.start();
        thread1.join();
        thread2.join();
        System.out.println("State of thread1: " + thread1.getState());
        System.out.println("State of thread2: " + thread2.getState());
    }
}

```

OUTPUT :-

```

PS C:\java> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\3ef1734a098faed0b01e94968e1146d1\redhat.java\jdt_ws\java_3e0290da\bin' 'multiplication5'
Multiplication table for 5
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
State of thread1: TERMINATED
State of thread2: TERMINATED
PS C:\java>

```