Name: Atharva Desai   Sap ID- 60004210202   Div/Batch-C/C1   Branch- Comps

JAVA EXPERIMENTS

Experiment 1

Aim:

To implement Java program structures and Simple programs.

Q1) Printing Hello

Code:

```
class Hello {
    public static void main(String[] args) {
        System.out.println("Hello!");
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac Hello.java
PS C:\Users\Abhishek\Documents\java_pracs> java Hello
Hello!
```

Q2) Count Number of Digits

Code:

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

        int noOfDigits = 0;
        System.out.println("Enter a Number");
        int num = Integer.parseInt(args[0]);
        while (num > 0) {
            noOfDigits++;
            num = num / 10;
        }
        System.out.println("Number of Digits: " + noOfDigits);
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac CountDigits.java
PS C:\Users\Abhishek\Documents\java_pracs> java CountDigits 12462
Enter a Number
Number of Digits: 5
```

Experiment 2

Aim: To implement Java control statements and loops

Q1) Roots of a Quadratic Equation

Code :

```
import java.util.Scanner;

class Quadratic {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter The Coefficients of the Quadratic Equation (From Higher powers to constants)");
        double a = ob.nextDouble();
        double b = ob.nextDouble();
        double c = ob.nextDouble();
        ob.close();
        System.out.println("Quadratic Equation: " + a + "x^2 + " + "(" + b + "x) +" + "(" + c + ")");
        double[] roots = new double[3];
        roots = quadRoots(a, b, c);
        if (roots[2] == 1)
            System.out.println("Root1: " + (roots[0] - roots[1]) + "\nRoot2: " + (roots[1] + roots[0]));
        else
            System.out.println("Root1: " + roots[0] + "+" + roots[1] + "i\nRoot2: " + roots[0] + "+" + roots[1] + "i");

    }

    public static double[] quadRoots(double a, double b, double c) {
        double[] roots = new double[3];
        double sqrtTerm = ((b * b) - (4 * a * c)) >= 0.0 ? Math.sqrt(((b * b) - (4 * a * c)))
                : Math.sqrt((-1 * (((b * b) - (4 * a * c)))));

        roots[2] = ((b * b) - (4 * a * c)) >= 0.0 ? 1 : 0;
        roots[0] = ((-1 * b)) / (2 * a);
        roots[1] = sqrtTerm / (2 * a);
        return roots;
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac Quadratic.java
PS C:\Users\Abhishek\Documents\java_pracs> java Quadratic
Enter The Coefficients of the Quadratic Equation (From Higher powers to constants)
1
4
5
Quadratic Equation: 1.0x^2 + (4.0x) +(5.0)
Root1: -2.0+1.0i
Root2: -2.0+1.0i
```

Q3) Menu Driven Calculator

Code:

```
import java.util.Scanner;

class Menu {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        double num1, num2;
        System.out.print("Enter first number:");
        num1 = ob.nextDouble();
        System.out.print("Enter second number:");
        num2 = ob.nextDouble();

        System.out.print("Enter an operator (+, -, *, /): ");
        char operator = ob.next().charAt(0);
        double output = 0.0;

        switch (operator) {
            case '+':
                output = num1 + num2;
                break;

            case '-':
                output = num1 - num2;
                break;
            case '*':
                output = num1 * num2;
                break;

            case '/':
                output = num1 / num2;
                break;

            default:
                System.out.printf("You have entered the wrong operator");

        }

        System.out.println(num1 + " " + operator + " " + num2 + " = " + output);
        ob.close();
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac Menu.java
PS C:\Users\Abhishek\Documents\java_pracs> java Menu Enter
first number:2
Enter second number:4
Enter an operator (+, -, *, /): *
2.0 * 4.0 = 8.0
```

Q3) Default Values of Primitive Data Types

Code :

```
class Default {
    public static int i;
    public static double d;
    public static short s;
    public static long l;
    public static float f;
    public static char c;
    public static boolean b;
    public static byte bi;

    public static void main(String[] args) {
        System.out.println("Default Value of Integer: " + i);
        System.out.println("Default Value of Double: " + d);
        System.out.println("Default Value of Float: " + f);
        System.out.println("Default Value of Char: " + c);
        System.out.println("Default Value of Boolean: " + b);
        System.out.println("Default Value of Long: " + l);
        System.out.println("Default Value of Short: " + s);
        System.out.println("Default Value of Byte: " + bi);

    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac Default.java
PS C:\Users\Abhishek\Documents\java_pracs> java Default
Default Value of Integer: 0
Default Value of Double: 0.0
Default Value of Float: 0.0
Default Value of Char:
Default Value of Boolean: false
Default Value of Long: 0
Default Value of Short: 0
Default Value of Byte: 0
```

Q4) Print Odd and Prime Numbers in Given Range

Code:

```
import java.util.Scanner;

class oddAndPrime {
    public static void main(String[] args) {
        int count = 0;
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a Range of Numbers");
        int beg = ob.nextInt();
        int end = ob.nextInt();
        System.out.println("Odd Numbers in Range");
        for (int i = beg; i < end; i++) {
            if (i % 2 == 1)
                System.out.println(i);
        }

        System.out.println("Prime Numbers in Range");
        for (int j = beg; j < end; j++) {
            for (int k = 2; k < j; k++) {
                if (j % k == 0) {
                    count++;
                }
            }
            if (count == 0)
                System.out.println(j);
            count = 0;
        }
        ob.close();

    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac oddAndPrime.java
PS C:\Users\Abhishek\Documents\java_pracs> java oddAndPrime Enter
a Range of Numbers
2
20
Odd Numbers in Range
3
5
7
9
11
13
15
17
19
Prime Numbers in Range
2
3
5
7
11
13
17
19
```

Q5) Print Pattern

Code:

```
class Pattern {
    public static void main(String[] args) {
        System.out.println("Number Pattern: ");
        for (int i = 0; i <= 7; i++) {
            if (i % 2 == 1) {
                for (int j = 1; j <= i; j++) {
                    System.out.print(j + " ");
                }
                System.out.println("");
            } else {
                for (int k = i; k > 0; k--) {
                    System.out.print(k + " ");
                }
                System.out.println("");
            }

        }
        int a = 0;
        Int char_inc = 3;
        System.out.println("Letter Pattern:");
        for (int h = 0; h < 4; h++) {
            for (int g = 3; g > h; g--) {
                System.out.print(" ");
            }
            for (int l = 0; l <= h; l++) {
                System.out.print((char) (a + 65));
                a--;
            }
            System.out.println("");
             a+=char_inc;
             char_inc+=2;
        }

    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\java_pracs> javac Pattern.java
PS C:\Users\Abhishek\Documents\java_pracs> java Pattern
Number Pattern:

 1
2 1
 1 2 3
4 3 2 1
 1 2 3 4 5
```

```
6 5 4 3 2 1
1 2 3 4 5 6 7
Letter Pattern:
   A
  CB
 FED
JIHG
```

Experiment 3:

Aim:

To implement Arrays

(i) WAP to find whether the entered 4 digit number is a vampire or not.
Combination of
digits from this number form 2 digit number. When they are multiplied by each other we get the original number. (1260=21*60, 1395=15*93, 1530=30*51)

Code:

```
package experiment_3;

import java.util.Scanner;

class vampire {

    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a 4-digit Number");
        int number = ob.nextInt();
        ob.close();
        int copy = number;
        int digits[] = new int[4];
        for (int i = 0; i < 4; i++) {
            digits[i] = number % 10;
            number /= 10;
        }
        for (int j = 0; j < 4; j++) {
            System.out.println(digits[j]);
        }

        String result = isVampire(digits, copy) ? " a Vampire Number" : " not a Vampire Number";
        System.out.println(copy + " is" + result);
    }
    public static boolean isVampire(int[] digits, int number) {
        // System.out.println("You Have called isVampire");

        int compare = 0;
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                for (int k = 0; k < 4; k++) {
                    for (int l = 0; l < 4; l++) {

                        if (i != j && j != k && k != l && l != i && j != l && i != k) {
                            compare = ((digits[i] * 10 + digits[j]) * (digits[k] * 10 + digits[l]));
                            // System.out.println(compare);
                            if (compare == number) {
                                return true;

                            } // if
                        } // outer if
                    } // l
                } // k

            } // j
        } // i
        return false;
    }

}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_3> javac vampire.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_3> java vampire.java
Enter a 4-digit Number
 1260
 1260 is a Vampire Number
```

(ii) WAP to display the following using irregular arrays

1
2 3
4 5 6

Code:

```
package experiment_3;
```

```java
import java.util.Scanner;

class Irregular {
    public static void main(String args[]) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter number of rows: ");
        int n = ob.nextInt();
        int[][] arr = new int[n][];
        for (int i = 0; i < n; i++) {
            arr[i] = new int[i + 1];
        }

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < arr[i].length; j++) {
                System.out.print("Enter value:");
                arr[i][j] = ob.nextInt();
            }
        }

        ob.close();

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < arr[i].length; j++) {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println("");
        }
    }

}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_3> javac .\Irregular.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_3> java Irregular.java
Enter number of rows:
4
Enter value:1
Enter value:2
Enter value:3
Enter value:7
Enter value:2
Enter value:6
Enter value:3
Enter value:6
Enter value:29
Enter value:08
 1
2 3
```

```
7 2 6
3 6 29 8
```

(iii) Write a program that queries the number of rows and columns from the user as student and their marks. Read the data and display in a tabular format

Code:

```
package experiment_3;

import java.util.*;

public class exp3_3 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the number of students:");
        int n_students = s.nextInt();
        System.out.println("Enter the number of subjects:");
        int n_subjects = s.nextInt();
        int i, j;
        int[][] arrayforeachstudent = new int[n_students][n_subjects];

        System.out.println("Enter the marks for all students:");
        for (i = 0; i < n_students; i++) {

            for (j = 0; j < n_subjects; j++) {
                arrayforeachstudent[i][j] = s.nextInt();

            }
        }
        // System.out.println(Arrays.deepToString(arrayforeachstudent));

        // printing values
        System.out.println("printing values\n");
        int allthemarks = 0;
        for (i = 0; i < n_students; i++) {
            int totalmarksperstudent = 0;
            for (j = 0; j < n_subjects; j++) {
                totalmarksperstudent += arrayforeachstudent[i][j];
                System.out.print(arrayforeachstudent[i][j] + " ");
            }
            System.out.print(" | " + totalmarksperstudent + "\n");
            allthemarks += totalmarksperstudent;
        }
        // array for storing marks per subject
        int[] indisubmarks = new int[n_subjects];
        System.out.println("------------------------------------------------------------");
        // finding total marks per subject
        for (j = 0; j < n_subjects; j++) {
            indisubmarks[j] = 0;
            for (i = 0; i < n_students; i++) {
                indisubmarks[j] += arrayforeachstudent[i][j];
            }
            // printing last row for total individual subject marks

            System.out.print(indisubmarks[j] + " ");

        }
        System.out.print(" | " + allthemarks);

        s.close();
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_3> javac exp3_3.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_3> java exp3_3.java
Enter the number of students:
2
Enter the number of subjects:
3
Enter the marks for all students:
12
89
34
53
24
56
printing values

12 89 34 | 135
53 24 56 | 133
------------------------------------------------------------
65 113 90 | 268
```

Experiment 4:

Aim:

To implement vectors

(i) WAP that accepts a shopping list of items and performs the following operations: Add an item at a specified location, delete an item in the list, and print the contents of the vector

Code:

```
package experiment_4;

import java.util.*;

class exp4_1 {
    static void printmenu() {
        System.out.println("1. Add an item to a speific position");
        System.out.println("2. Delete an item from a speific position");
        System.out.println("3. Show list");
        System.out.println("4. EXIT ");
    }

    public static void main(String[] args) {
        Vector<String> shoppingList = new Vector<String>();
        Scanner s = new Scanner(System.in);

        for (int i = 0; i < args.length; i++) {
            shoppingList.add(args[i]);
        }

        printmenu();

        System.out.println("Enter the command number you want to execute: ");

        int k = s.nextInt();

        while (k != 4) {
            switch (k) {
                case 1:
                    System.out.println("Enter the item to be added:");
                    String tobeadded = s.next();
                    System.out.println("Enter the position to be added at");
                    int i = s.nextInt();
                    shoppingList.insertElementAt(tobeadded, (i - 1));
                    System.out.println("Item added!");
                    printmenu();
                    System.out.println("Enter next command number");
                    k = s.nextInt();
                    break;

                case 2:

                    System.out.println("Enter the position of the item to be deleted:");
                    int p = s.nextInt();
                    shoppingList.remove(p - 1);
                    System.out.println("Item removed!");
                    printmenu();
                    System.out.println("Enter next command number");
                    k = s.nextInt();
                    break;

                case 3:
                    System.out.println("Shopping list is... \n");

                    for (int l = 0; l < shoppingList.size(); l++) {
                        System.out.println((l + 1) + ")" + shoppingList.elementAt(l));
                    }
                    System.out.println("\n");

                    printmenu();
                    System.out.println("Enter next command number");
                    k = s.nextInt();
                    break;

                default:
                    System.out.println("Please enter a valid command number:");
                    k = s.nextInt();
                    break;

            }
        }
        // System.out.print(shoppingList);
        s.close();
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_4> javac 4_1.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_4> java 4_1.java
1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter the command number you want to execute:

 1
Enter the item to be added:
Aditya
Enter the position to be added at
 1
 Item added!
 1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter next command number
 1
Enter the item to be added:
Micheal
Enter the position to be added at
2
 Item added!
 1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter next command number
 1
Enter the item to be added:
Jason
Enter the position to be added at
3
 Item added!
 1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter next command number
3
 Shopping list is...
```

```
1)Aditya
2)Micheal
3)Jason


1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter next command number
2
Enter the position of the item to be deleted:
2
Item removed!
1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter next command number
3
Shopping list is...

1)Aditya
2)Jason


1. Add an item to a speific position
2. Delete an item from a speific position
3. Show list
4. EXIT
Enter next command number
4
```

(ii) Write a java programs to find frequency of an element in the given Vector array.

Code:

```
package experiment_4;
import java.util.*;

class exp4_2 {
   public static void main(String[] args) {
      // filling vector array
      Scanner s = new Scanner(System.in);
      Vector<Integer> vec = new Vector<Integer>();
```

```
        System.out.println("How many elements are in the vector?");
        int n = s.nextInt();
        for (int i = 0; i < n; i++) {
            System.out.println("Enter element " + (i + 1));
            int elementobeadded = s.nextInt();
            vec.add(i, elementobeadded);
        }
        System.out.println("Which integers frequency do you want?");
        int k = s.nextInt();
        int count = 0;
        for (int i = 0; i < n; i++) {
            if (vec.get(i) == k) {
                count++;
            }
        }
        System.out.println("Frequency of " + k + " is " + count);

        // System.out.print(vec);
        s.close();

    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_4> javac 4_2.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_4> java 4_2.java
How many elements are in the vector?
5
Enter element 1
1
Enter element 2
2
Enter element 3
1
Enter element 4
4
Enter element 5
1
Which integers frequency do you want?
1
Frequency of 1 is 3
```

Experiment 5:

Aim:

To implement Strings

(i) WAP to check if 2 strings are Meta strings or not. Meta strings are the strings which can be made equal by exactly one swap in any of the strings. An equal string is not considered here as Meta strings.
Example: str1 = "geeks”,
str2 = "keegs"
By just swapping 'k' and 'g' in any of string, both will become the same.
Example: str1 = "Converse", str2 = "Conserve"
By just swapping 'v' and’s’ in any of string, both will become the same.

Code:

```
package experiment_5;

import java.util.Scanner;

public class MetaString {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter two strings to check if they are Meta:");
        String s1 = s.next();
        String s2 = s.next();
        char[] charstobeswapped = new char[4];

        boolean flag = false;
        if( s1.length() != s2.length() ){
            flag = false;
        }else{
            int j = 0;
            s1 = s1.toLowerCase();
            s2 = s2.toLowerCase();
            int noOfDifferentLetters = 0;
            for(int i = 0 ; i <s1.length();i++){
                if( s1.charAt(i) != s2.charAt(i)){
                    noOfDifferentLetters++;
                    if(noOfDifferentLetters > 2){
                        flag = false;
                        break;
                    }
                    charstobeswapped[j] = s1.charAt(i);
                    charstobeswapped[j+1] = s2.charAt(i);
                    j = j+2 ;
                }
            }
        }
        if(charstobeswapped[0] == charstobeswapped[3] && charstobeswapped[1] == charstobeswapped[2]){
            flag = true;
        }
        s.close();
        //final statement

        if(flag == true){
            System.out.println("The two strings are Meta");
        }else{
            System.out.println("The two strings are Not Meta");
        }

    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_5> javac MetaString.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_5> java MetaString.java
Enter two strings to check if they are Meta:
geeks
keegs
The two strings are Meta
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_5> java MetaString.java
Enter two strings to check if they are Meta:
hello
gelly
The two strings are Not Meta
```

(ii) Write a java program to count the number of alphabets, digits, special symbols, blank spaces, and words from the given sentence. Also, count the number of vowels and consonants.

Code:

```
package experiment_5;

import java.util.Scanner;
class StringBreak {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a String");
        String s = ob.nextLine();
        ob.close();
        int[] result = checkCount(s);
        System.out.println("Number of Alphabets: " + result[0]);
        System.out.println("Number of Consonants: " + result[1]);
        System.out.println("Number of Vowels: " + result[2]);
        System.out.println("Number of numbers: " + result[3]);
        System.out.println("Number of Blank Spacses: " + result[4]);
        System.out.println("Number of Special Characetrs: " + result[5]);
    }

    public static int[] checkCount(String s) {
        s = s.toLowerCase();
        int[] count = new int[6];
        for (int i = 0; i < s.length(); i++) {
            if (Character.isLetter(s.charAt(i))) {
                count[0]++;
                if ("aeiou".indexOf(Character.toString(s.charAt(i))) == -1)
                    count[2]++;
                else
                    count[1]++;
            } else if (Character.isDigit(s.charAt(i)))
                count[3]++;
            else if (Character.isSpaceChar(s.charAt(i)))
                count[4]++;
            else
                count[5]++;
        }
        return count;
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_5> javac StringBreak.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_5> java StringBreak.java
Enter a String
Hello My Name is @d!tya
Number of Alphabets: 17
Number of Consonants: 6
Number of Vowels: 11
Number of numbers: 0
Number of Blank Spacses: 4
Number of Special Characetrs: 2
```

Experiment 6

Aim:

To implement functions, recursive functions, and overloading

(i) WAP to display area of square and rectangle using the concept of overloaded functions

Code:

```
package experiment_6;

class Overloading {

    public void rectArea(double length, double breadth) {
        double area = length * breadth;
        System.out.println("Area of Rectangle (double): " + area);
    }

    public void rectArea(double side) {
        double area = side * side;
        System.out.println("Area of Square (double): " + area);
    }

    public void rectArea(int length, int breadth) {
        int area = length * breadth;
        System.out.println("Area of Rectangle (int): " + area);
    }

    public static void main(String[] args) {
        Overloading o = new Overloading();
        o.rectArea(1, 5);
        o.rectArea(4);
        o.rectArea(2.5, 3.5);
    }

}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Overloading.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Overloading.java
Area of Rectangle (int): 5
Area of Square (double): 16.0
Area of Rectangle (double): 8.75
```

(ii) Write a menu-driven program to implement recursive functions for the following tasks.

a) To find GCD and LCM

Code:

```
package experiment_6;

import java.util.Scanner;

class Gcd_Lcm {

    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter 2 numbers");
        int a = ob.nextInt();
        int b = ob.nextInt();
        ob.close();
        System.out.println("GCD: " + Calc_GCD_LCM.gcd(a, b));
        System.out.println("LCM: " + Calc_GCD_LCM.lcm(a, b));

    }

}

class Calc_GCD_LCM {

    // using Euclidian Algorithm
    public static int gcd(int a, int b) {
        if (b != 0)
            return gcd(b, a % b);
        return a;
    }

    public static int lcm(int a, int b) {
        return (a * b) / gcd(a, b);
    }}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Gcd_Lcm.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Gcd_Lcm.java
Enter 2 numbers
 12
72
```

```
GCD: 12
LCM: 72
```

b) To find X^Y

Code:

```
package experiment_6;

import java.util.*;

class Power {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a base and exponent");
        double base = ob.nextDouble();
        double exp = ob.nextDouble();
        ob.close();
        double ans = power(base, exp);
        System.out.println("Result: " + ans);
    }

    public static double power(double base, double exponent) {
        if (exponent == 1)
            return base;
        else
            return base * power(base, exponent - 1);
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Power.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Power.java
Enter a base and exponent
2
6
Result: 64.0
```

c) To print n Fibonacci numbers

Code:

```
package experiment_6;

import java.util.Scanner;
```

```
public class Fibonacci {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter which term of Fibonacci is required");
        int n = ob.nextInt();
        ob.close();
        System.out.println("Fibonacci: " + Function.fib(n - 1));
    }

}

class Function {
    public static int fib(int n) {
        if (n == 1 || n == 0)
            return n;
        else
            return fib(n - 2) + fib(n - 1);
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Power.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Power.java
Enter a base and exponent
6
Result: 64.0
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Fibonacci.java
Enter which term of Fibonacci is required
6
Fibonacci: 5
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Fibonacci.java
Enter which term of Fibonacci is required
1
Fibonacci: 0
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Fibonacci.java
Enter which term of Fibonacci is required
3
Fibonacci: 1
```

d) To find the reverse of a number

Code:

```
package experiment_6;

import java.util.*;

public class Reverse {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a Number:");
        int num = ob.nextInt();
        ob.close();
        System.out.println(ReverseFunc.reverse(num));
    }
}

class ReverseFunc {
    static int rev = 0;
    static int mult_fact = 1;

    public static int reverse(int n) {
        if (n > 0) {
            reverse(n / 10);
            rev += (n % 10) * mult_fact;
            mult_fact *= 10;
        }
        return rev;
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Reverse.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Reverse.java
Enter a Number:
 12575
57521
```

e) To 1+2+3+4+.......+ (n-1)+n

Code:

```
package experiment_6;

import java.util.Scanner;

public class Series {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a Number:");
        int num = ob.nextInt();
        ob.close();
        System.out.println(SeriesSum.sum(num));
    }
}
```

```
class SeriesSum {
    public static long sum(int n) {
        if (n == 1)
            return 1;
        return n + sum(n - 1);
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Series.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Series.java
Enter a Number:
5
15
```

f) Calculate the sum of digits of a number

Code:

```
package experiment_6;

import java.util.*;

public class Sum {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter a Number:");
        int num = ob.nextInt();
        ob.close();
        System.out.println("Sum of Digigts: " + SumDig.sumOfDig(num));
    }
}

class SumDig {
    public static int sumOfDig(int n) {
        if (n == 1 || n == 0)
            return n;
        return sumOfDig(n / 10) + (n % 10);
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> javac Sum.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_6> java Sum.java
Enter a Number:
 124353
Sum of Digits: 18
```

Experiment 7

Aim:

To implement Array of Objects

(i) WOOP to arrange the names of students in descending order of their total marks, input data consists of students' details such as names, ID.no, marks of maths, physics, chemistry. (Use an array of objects)

Code:

```
package experiment_7;

import java.util.Scanner;

public class ArrayOfObjects {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        int n;
        int i;
        int s_id;
        String name;
        long math_mark;
        long chem_mark;
        long phy_mark;
        System.out.println("Enter number of Students");
        n = ob.nextInt();
        Student[] objArr = new Student[n];
        System.out.println("Enter The Student Details");
        for (i = 0; i < n; i++) {
            ob.nextLine();
            System.out.print("Enter Student Name: ");
            name = ob.nextLine();
            System.out.println();
            System.out.print("Enter Student ID: ");
            s_id = ob.nextInt();
            System.out.println();
            System.out.print("Enter Math Marks of Student: ");
```

```
          math_mark = ob.nextInt();
          System.out.println();
          System.out.print("Enter Chem Marks of Student: ");
          chem_mark = ob.nextInt();
          System.out.println();
          System.out.print("Enter Physics Marks of Student: ");
          phy_mark = ob.nextInt();
          System.out.println();
          objArr[i] = new Student(name, s_id, phy_mark, chem_mark, math_mark);
          System.out.println("=============== Student Details Have Succussfully Been Entered
===================");

       }
       ob.close();
       System.out.println("=================== Student Details: ============================");
       for (i = 0; i < n; i++) {
          System.out.println("\nName = " + objArr[i].getName() + " Student ID: " + objArr[i].getID());
       }
       sort(objArr);
       System.out.println();
       System.out.println("==================== Sorted Students! =====================");
       for (i = 0; i < n; i++) {
          System.out.println("\nName = " + objArr[i].getName() + " Student ID: " + objArr[i].getID());
       }
    }

    static void sort(Student[] objArr) {
       for (int i = 0; i < objArr.length - 1; i++)
          for (int j = 0; j <= objArr.length - i - 2; j++)
             if (objArr[j].getTotalMarks() < objArr[j + 1].getTotalMarks()) {
                Student temp = objArr[j];
                objArr[j] = objArr[j + 1];
                objArr[j + 1] = temp;
             }
    }

}

class Student {

    // attributes | class instance variables
    private String name;
    private int s_ID;
    private long phy_marks, chem_marks, math_marks;
    private long total_marks;

    // constructor
    Student(String name, int s_ID, long phy_marks, long chem_marks, long math_marks) {
        this.name = name;
        this.math_marks = math_marks;
        this.phy_marks = phy_marks;
        this.chem_marks = chem_marks;
        this.s_ID = s_ID;
        this.total_marks = this.phy_marks + this.math_marks + this.chem_marks;
    }

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

    public int getID() {
        return this.s_ID;
    }

    public long getTotalMarks() {
        return this.total_marks;
    }

    public long getMathMarks() {
        return this.math_marks;
    }

    public long getPhyMarks() {
        return this.phy_marks;
    }

    public long getChemMarks() {
        return this.chem_marks;
    }

}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_7> java ArrayOfObjects.java
Enter number of Students
3
Enter The Student Details
Enter Student Name: Aditya

Enter Student ID: 006

Enter Math Marks of Student: 12

Enter Chem Marks of Student: 64

Enter Physics Marks of Student: 85
```

```
================ Student Details Have Succussfully Been Entered ====================
Enter Student Name: Abhshek

Enter Student ID: 032

Enter Math Marks of Student: 98

Enter Chem Marks of Student: 59

Enter Physics Marks of Student: 69

================ Student Details Have Succussfully Been Entered ====================
Enter Student Name: Micheal

Enter Student ID: 99

Enter Math Marks of Student: 9
Enter Chem Marks of Student: 99
Enter Physics Marks of Student: 99
================ Student Details Have Succussfully Been Entered ====================
==================== Student Details: =============================
Name = Aditya Student ID: 6
Name = Abhshek Student ID: 32
Name = Micheal Student ID: 99

===================== Sorted Students! ========================

Name = Abhshek Student ID: 32
Name = Micheal Student ID: 99
Name = Aditya Student ID: 6
```

Experiment 8

Aim:

To implement Constructors and overloading

.

(i) WAP to find the area of square and rectangle using overloaded constructor

Code:

```
package experiment_8;

class OverloadConstructor {

    private double length;
    private double breadth;
    private double area;

    public static void main(String[] args) {
        OverloadConstructor square = new OverloadConstructor(2.5);
        System.out.println("Area of Square: " + square.getArea());

        OverloadConstructor rect = new OverloadConstructor(2.5, 10.0);
        System.out.println("Area of Rectangle: " + rect.getArea());

    }

    public OverloadConstructor(double side) {
        this.length = side;
        this.breadth = side;
    }

    public OverloadConstructor(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double getLength() {
        return this.length;
    }

    public double getBreadth() {
        return this.breadth;
    }

    public double getArea() {
        this.area = this.length * this.breadth;
        return this.area;
    }}
```

Output:

```
Area of Square: 6.25
Area of Rectangle: 25.0
```

(ii) Create a Rectangle and Cube class that encapsulates the properties of a rectangle and cube i.e. Rectangle has default and parameterized constructor and area() method. Cube has default and parameterized constructor and volume() method.

They share no ancestor other than Object.

Implement a class Size with size() method. This method accepts a single reference argument z. If z refers to a Rectangle then size(z) returns its area and if z is a reference of Cube, then z returns its volume. If z refers to an object of any other class, then size(z) returns -1. Use main method in Size class to call size(z) method

Code:

```
package experiment_8;

class Rectangle {
   double length;
   double breadth;

   // Default constructor
   Rectangle() {
       length = 0;
      breadth = 0;
   }

   // Parameterized constructor
   Rectangle(double length, double breadth) {
       this.length = length;
       this.breadth = breadth;
   }

   // area method
   double area() {
       return length * breadth;
   }
}

class Cube {
   double side;
   // Default constructor
   Cube() {
       side = 0.0;
   }

   // Parameterized constructor
   Cube(double side) {
       this.side = side;
   }

   // volume method
   double volume() {
       return side * side * side;
   }
}

class Size {
   double calcSize(Object obj) {
       if (obj instanceof Rectangle)
           return ((Rectangle) obj).area();

       if (obj instanceof Cube)
           return ((Cube) obj).volume();

       return -1;

   }

   public static void main(String args[]) {
       Cube cube = new Cube(2);
       Rectangle rect = new Rectangle(2, 3);
       Size size = new Size();
       double isCube = size.calcSize(cube);
       System.out.println(isCube);
       double isRect = size.calcSize(rect);
       System.out.println(isRect);
       double defaultCase = size.calcSize(2);
   System.out.println(defaultCase); }
}
```

Output:

```
8.0
6.0
-1.0
```

Experiment 9

Aim:

To implement an abstract class

(i) Write an abstract class program to calculate the area of circle, rectangle, and triangle

Code:

```
package experiment_9;

class AbstractClass {
    public static void main(String[] args) {
        Rect r = new Rect(5.0, 6.0);
        Tri t = new Tri(3.0, 2.0);
        Circle c = new Circle(7.0);
        System.out.println("Area of Reactangle is: " + r.area());
        System.out.println("Area of Triangle is: " + t.area());
        System.out.println("Area of Circle is: " + c.area());

    }
}

abstract class Area {

    abstract double area();
}

class Rect extends Area {
    private double length;
    private double breadth;

    Rect(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;

    }

    double area() {
        return this.length * this.breadth;
    }
}

class Tri extends Area {
    private double height;
    private double base;
    Tri(double height, double base) {
        this.height = height;
        this.base = base;

    }

    double area() {
        return this.height * this.base * 0.5;
    }
}

class Circle extends Area {
    private double radius;

    Circle(double radius) {
        this.radius = radius;
    }
    double area() {
        return this.radius * this.radius * 3.14;
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_9> javac .\AbstractClass.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_9> java .\AbstractClass.java
Area of Rectangle is: 30.0
Area of Triangle is: 3.0
Area of Circle is: 153.86
```

Experiment 10

Aim:

To implement Inheritance, interfaces, and method overriding

(i) WAP to implement three classes namely Student, Test, and Result. Student class has member as rollno, Test class has members as sem1_marks and sem2_marks and The Result class has members as total. Create an interface named Sports that have a member score (). Derive Test class from Student and Result class has multiple inheritances from Test and Sports. Total is a formula based on sem1_marks, sem2_mark, and score.

Code:

```
package experiment_10;

import java.util.Scanner;

class Student {
    int rollNo;

    Student(int rollNo) {
        System.out.println("Student constructor was called");
        this.rollNo = rollNo;
    }

    void setRollno(int rollNo) {
        this.rollNo = rollNo;
    }

    int getRollNo() {
        return this.rollNo;
    }

}

class Test extends Student {
    int sem1Marks;
    int sem2Marks;

    Test(int sem1Marks, int sem2Marks, int rollNo) {
        super(rollNo);
        System.out.println("Test constructor was called");
        this.sem1Marks = sem1Marks;
        this.sem2Marks = sem2Marks;
    }

    void setSem1Marks(int sem1) {
        this.sem1Marks = sem1;
    }

    int getSem1Marks() {
        return this.sem1Marks;
    }

    void setSem2Marks(int sem2Marks) {
        this.sem2Marks = sem2Marks;
    }
    int getRollNo() {
        return this.sem2Marks;
    }
}

class Result extends Test implements Sports {
    int total;

    Scanner ob = new Scanner(System.in);

    Result(int sem1Marks, int sem2Marks, int rollNo) {

        super(sem1Marks, sem2Marks, rollNo);
        System.out.println("Result Constructor was called");
    }

    public void score() {
        this.total = (sem1Marks + sem2Marks) / 2 + score;
        System.out.println("The score of this student is (Average of sem1 and sem2 + sports score): " + this.total);
    }
    public static void main(String[] args) {
        Result res = new Result(80, 100, 6);
        res.score();
    }

}

interface Sports {
    int score = 49;
}
```

Output:

```
Student constructor was called
Test constructor was called
Result Constructor was called
The score of this student is (Average of sem1 and sem2 + sports score): 139
```

Experiment 11

Aim:

To implement package

(i) WAP to create a user-defined package & import the package in another program

```
package experiment_11.Package;

public class Volume {
    public double coneVol(double height, double radius) {
        return (0.333 * 3.14 * radius * radius * height);
    }

    public static void main(String[] args) {
        Volume v = new Volume();
        System.out.println("Volume: " + v.coneVol(5.0, 1.0));
    }
}

package experiment_11;

import experiment_11.Package.Volume;

class Cone {
    public static void main(String[] args) {
        Volume v = new Volume();
        System.out.println("Volume of cone with radius 5 and height 10 is: " + v.coneVol(10.0, 5.0));
    }
}
```

Output:

```
Volume of cone with radius 5 and height 10 is: 261.405
```

Experiment 12

Aim: To implement exceptions in Java

(i) Write a Java Program to input the data through command Line and Find out total valid and in-valid integers. (Hint: use exception handling)

Code:

```
package experiment_12;

import java.io.IOException;

public class IntegerException {
    public static void main(String[] args) throws IOException {
        int[] arr = new int[args.length];
        for (int i = 0; i < args.length; i++) {
            try {
                arr[i] = Integer.parseInt(args[i]);
                System.out.println("Valid");
            } catch (NumberFormatException e) {
                System.out.println("Invalid");
            }
        }
    }
}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_12> javac IntegerException.java PS
C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_12> java IntegerException.java 12 5 5 dc
oow 24 i
Valid
Valid
Valid
Invalid
Invalid
Valid
Invalid
```

(ii) Write a Java Program to calculate the Result. Result should consist of name, seatno, date, center number and marks of semester three exam. Create a User Defined Exception class MarksOutOfBoundsException, If Entered marks of any subject is greater than 100 or less than 0, and then program should create a user defined Exception of type MarksOutOfBoundsException and must have a provision to
handle it.

Code:

```
package experiment_12;

import java.util.Scanner;

class Result {

    private int marks;
    private int dataCenter;
    private int seatNo;
    private String name;
    private String Date;

    public Result(int marks, int dataCenter, int seatNo, String name, String Date) {
        this.marks = marks;
        this.dataCenter = dataCenter;
        this.seatNo = seatNo;
        this.name = name;
        this.Date = Date;
    }

    int getMarks() {
        return marks;
    }

    int getSeatNo() {
        return seatNo;
    }

    int getDataCenter() {
        return dataCenter;
    }

    String getName() {
        return name;
    }
    String getDate() {
        return Date;
    }

    void ValidateMarks(int marks) throws MarksOutOfBoundsException {
        if (this.marks > 100 || this.marks < 0)
            throw new MarksOutOfBoundsException("Marks out Of Bounds");
    }

    public static void main(String[] args) {

    }
}

class MarksOutOfBoundsException extends Exception {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public MarksOutOfBoundsException(String message) {
        super(message);
    }
}

class CalcResult {
    public static void main(String[] args) {
        Scanner ob = new Scanner(System.in);
        System.out.println("Enter Marks, Data centre number, Seat number, Name and Date");
        int marks = ob.nextInt();
        int dataCenterNo = ob.nextInt();
        int seatNo = ob.nextInt();
        String name = ob.next();
        String date = ob.next();
        ob.close();
        Result r = new Result(marks, dataCenterNo, seatNo, name, date);
        try {
            r.ValidateMarks(marks);
            System.out.println(r.getMarks());
        } catch (MarksOutOfBoundsException e) {
            System.out.println("Exception caught");
            System.out.println(e.getMessage());
        }

    }
}
```

Output:

```
Enter Marks, Data centre number, Seat number, Name and Date
75
6
Aditya
October
```

```
Enter Marks, Data centre number, Seat number, Name and Date
 10000
24
24
56
Aditya
Exception caught
Marks out Of Bounds
```

Experiment 13

Aim:

To implement Multi-threading

(i) Write java program to print Table of Five, Seven and Thirteen using Multithreading (Use Thread class for the implementation).
Also print the total time taken by each thread for the execution.

Code:

```
package experiment_13;

public class Tables {
    public static void main(String[] args) {
        Five f = new Five();
        Seven s = new Seven();
        Thirteen t = new Thirteen();
        f.start();
        s.start();
        t.start();

    }
}

class Seven extends Thread {
    public void run() {
        long t1 = System.currentTimeMillis();
        for (int i = 1; i <= 10; i++) {
            System.out.println("7 x " + i + " = " + (7 * i));
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
        long t2 = System.currentTimeMillis();
        System.out.println("Time Taken by table of 7: " + (t2 - t1));
    }
}

class Five extends Thread {
    public void run() {
        long t1 = System.currentTimeMillis();
        for (int i = 1; i <= 10; i++) {
            System.out.println("5 x " + i + " = " + (5 * i));
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
        long t2 = System.currentTimeMillis();
        System.out.println("Time Taken by table of 5: " + (t2 - t1));
    }
}

class Thirteen extends Thread {
    public void run() {
        long t1 = System.currentTimeMillis();
        for (int i = 1; i <= 10; i++) {
            System.out.println("13 x " + i + " = " + (13 * i));
            try {
                Thread.sleep(1000);
            } catch (Exception e) {
            }
        }
        long t2 = System.currentTimeMillis();
        System.out.println("Time Taken by table of 13: " + (t2 - t1));
    }

}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_13> javac Tables.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_13> java Tables.java
13 x 1 = 13
5 x 1 = 5
7 x 1 = 7
13 x 2 = 26
5 x 2 = 10
7 x 2 = 14
13 x 3 = 39
7 x 3 = 21
```

```
5 x 3 = 15
13 x 4 = 52
7 x 4 = 28
5 x 4 = 20
13 x 5 = 65
7 x 5 = 35
5 x 5 = 25
13 x 6 = 78
7 x 6 = 42
5 x 6 = 30
13 x 7 = 91
7 x 7 = 49
5 x 7 = 35
13 x 8 = 104
7 x 8 = 56
5 x 8 = 40
13 x 9 = 117
7 x 9 = 63
5 x 9 = 45
13 x 10 = 130
7 x 10 = 70
5 x 10 = 50
Time Taken by table of 13: 10104
Time Taken by table of 7: 10135
Time Taken by table of 5: 10139
```

(ii) Write java program to implement the concept of Thread Synchronization

Code:

```
package experiment_13;

public class Synchronization {
    public static void main(String[] args) {
        Flight f = new Flight(2);
        Thread t1 = new Thread(f);
        t1.setName("Aditya");
        Thread t2 = new Thread(f);
        t2.setName("Abhishek");
        Thread t3 = new Thread(f);
        t3.setName("Micheal Myers");
        t1.start();
        t2.start();
        t3.start();
    }
}
```

```
class Flight extends Thread {
    int vacant = 5, required;

    Flight(int required) {
        this.required = required;
    }

    public synchronized void run() {
        if (vacant >= required) {
            System.out.println(required + " Tickets Have Been Booked For: " + Thread.currentThread().getName() + "!");
            try {
                Thread.sleep(100);
            } catch (Exception e) {
            }

            vacant -= required;
        } else {
            System.out.println("Sorry " + Thread.currentThread().getName() + ", But The Flight is Booked! (Only "
                    + vacant + " Seat(s) Available)");
        }
    }

}
```

Output:

```
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_13> javac Synchronization.java
PS C:\Users\Abhishek\Documents\dsa\DataStructuers\Practicals\java\experiment_13> java Synchronization.java
2 Tickets Have Been Booked For: Aditya!
2 Tickets Have Been Booked For: Micheal Myers!
Sorry Abhishek, But The Flight is Booked! (Only 1 Seat(s) Available)
```

Experiment 14

Aim:

To Implement Applets

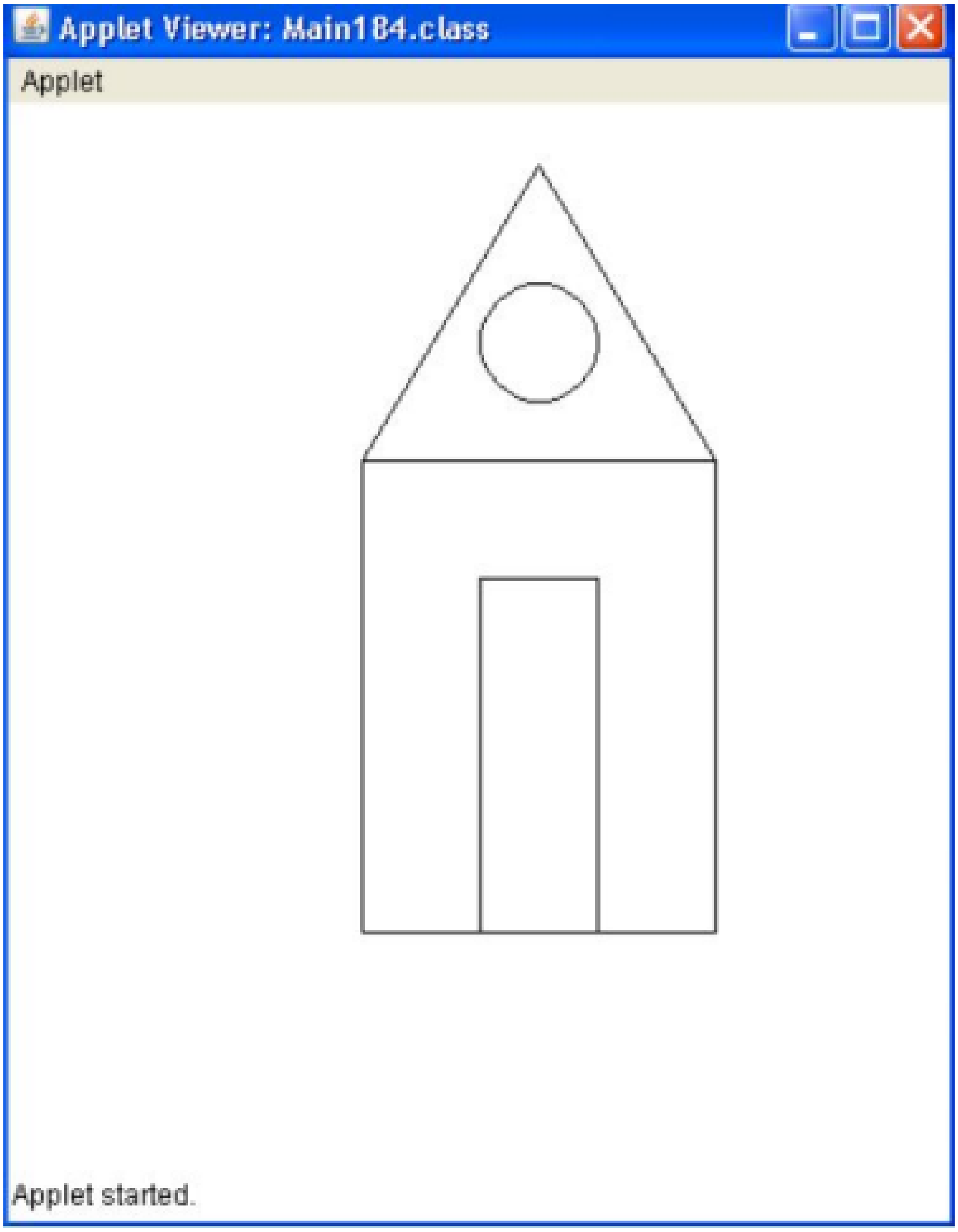
(i)Write java program to draw the house on an applet.

Code:

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


public class Main184 extends Applet
{
```

```
/*
<applet code="Main184.class" width=400 height=450></applet>
*/
public void paint(Graphics gp)
 { int [] x = {150, 300, 225};
 int [] y = {150, 150, 25};
 gp.drawRect(150, 150, 150, 200); //House
 gp.drawRect(200, 200, 50, 150); // Door
 gp.drawOval(200, 75, 50, 50); // Skylight
 gp.drawPolygon(x, y, 3); // Roof
 }
 }
```

Output:

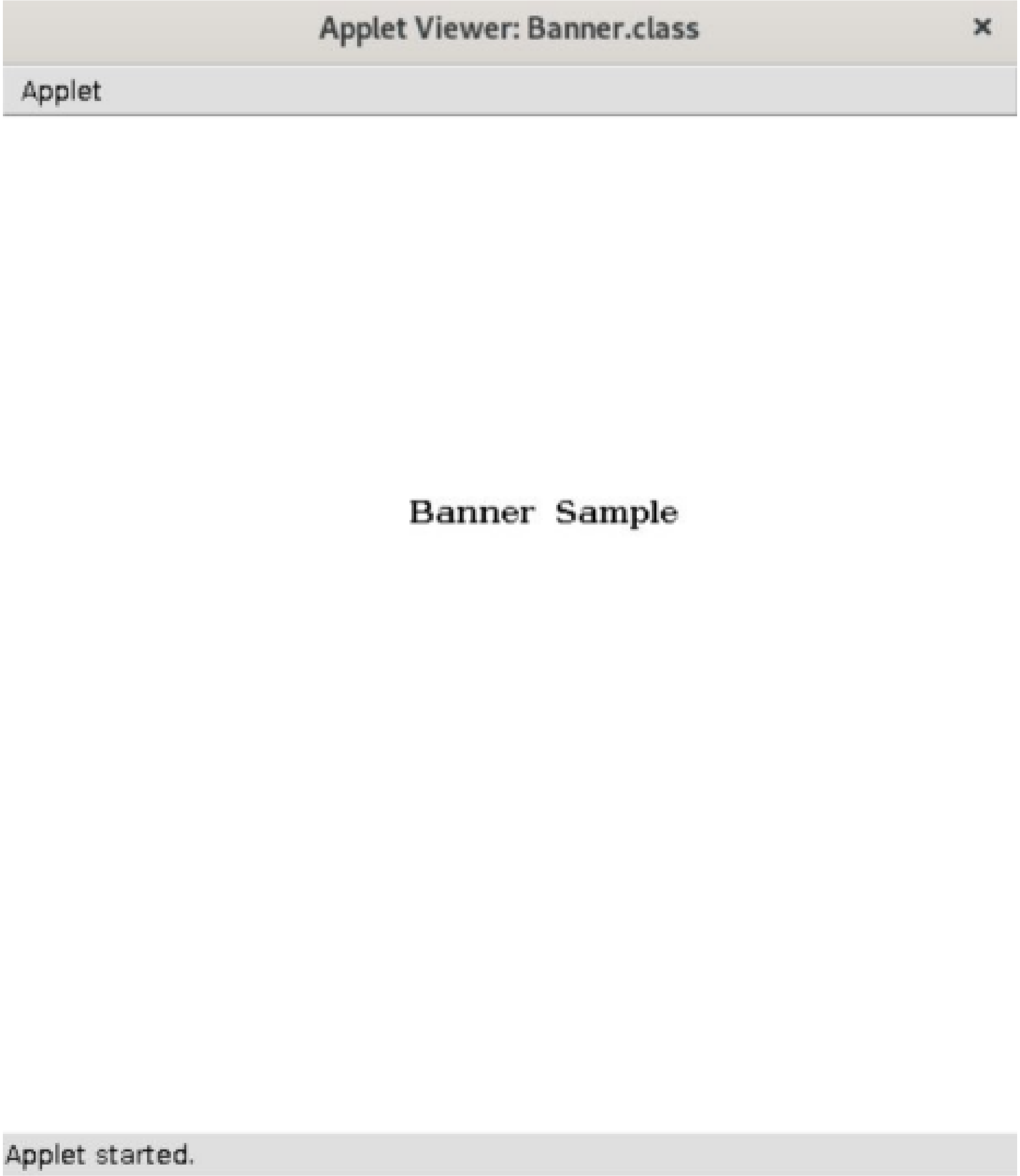


(ii) Write java program to create an advertisement banner on an applet using multithreading

Code:

```
import java.applet.*;
import java.awt.*;
public class Banner extends Applet implements Runnable
{
  String text = " Sample Banner ";
  Thread t;
  //Initialize the applet
  public void init()
  {
    setBackground(Color.white);
  }
  //Function to start the thread
  public void start()
  {
         t = new Thread(this);
         t.start();
  }
  //Function to execute the thread
  public void run()
  {
    while(true)
         {
           try
            {
              repaint();
                  //Delay each thread by 1000ms or 1 seconds
          Thread.sleep(1000);
                  //Shift the first character of banner text to the last postion
          text = text.substring(1) + text.charAt(0);
            }
           catch(Exception e)
            {
            }
         }
  }
  //Function to draw text
  public void paint(Graphics g)
  {
         g.setFont(new Font("TimesRoman",Font.BOLD,15));
         g.drawString(text,200,200);
  }
}
```

Output:



Banner Sample

Applet started.

Experiment 15

Aim:

Designing Graphical User Interfaces in Java using AWT and Event handling (i)

Write java program to create a registration form using AWT.

Code:

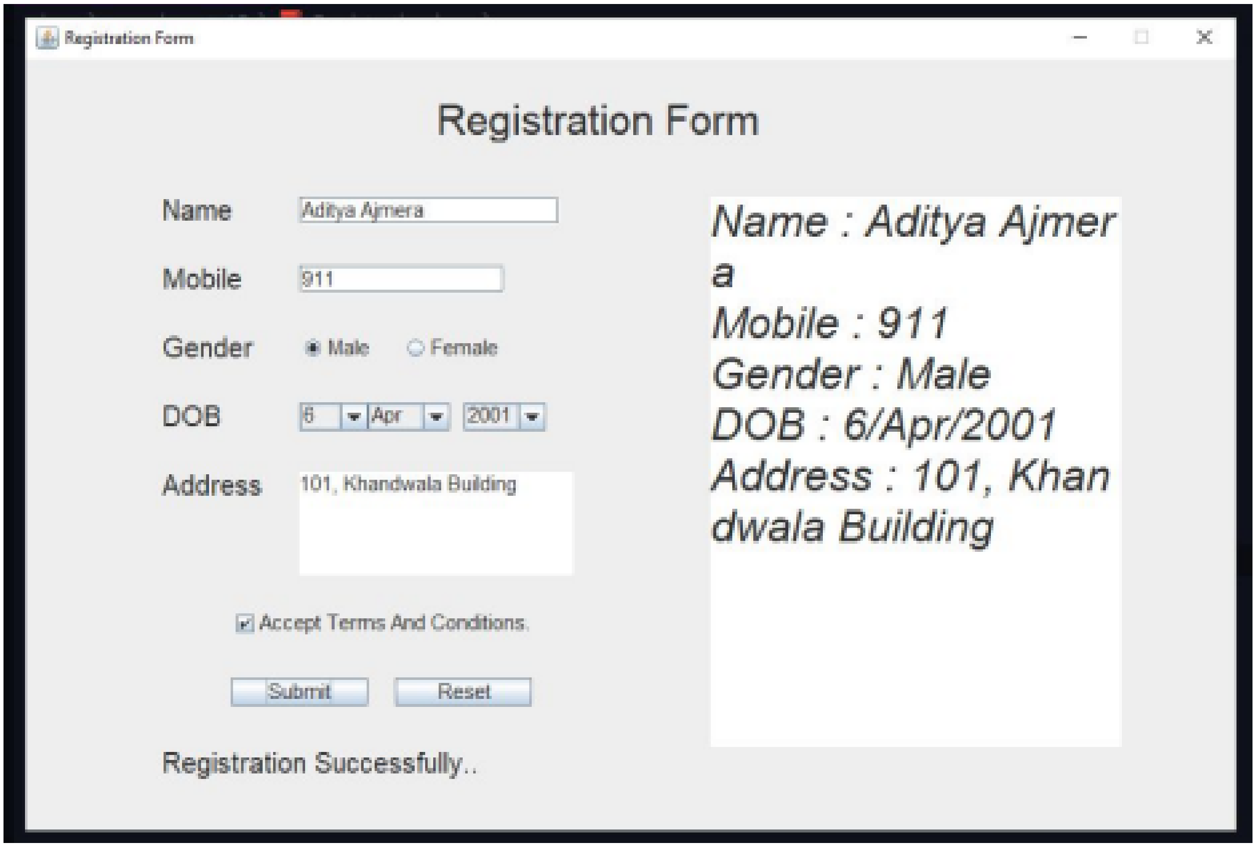
```
package experiment_15;

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

class MyFrame extends JFrame implements ActionListener {

    /**
     *
     */
    private static final long serialVersionUID = 1L;
    // Components of the Form
    private Container c;
    private JLabel title;
    private JLabel name;
    private JTextField tname;
    private JLabel mno;
    private JTextField tmno;
    private JLabel gender;
    private JRadioButton male;
    private JRadioButton female;
    private ButtonGroup gengp;
    private JLabel dob;
    private JComboBox date;
    private JComboBox month;
    private JComboBox year;
    private JLabel add;
    private JTextArea tadd;
    private JCheckBox term;
    private JButton sub;
    private JButton reset;
    private JTextArea tout;
    private JLabel res;
    private JTextArea resadd;
private String dates[] = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16",
            "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31" };
private String months[] = { "Jan", "feb", "Mar", "Apr", "May", "Jun", "July", "Aug", "Sup", "Oct", "Nov", "Dec" };
private String years[] = { "1995", "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "2004",
"2005",
        "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017", "2018",
        "2019" };

// constructor, to initialize the components
// with default values.
public MyFrame() {
    setTitle("Registration Form");
    setBounds(300, 90, 900, 600);
    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setResizable(false);

    c = getContentPane();
    c.setLayout(null);

    title = new JLabel("Registration Form");
    title.setFont(new Font("Arial", Font.PLAIN, 30));
    title.setSize(300, 30);
    title.setLocation(300, 30);
    c.add(title);

    name = new JLabel("Name");
    name.setFont(new Font("Arial", Font.PLAIN, 20));
    name.setSize(100, 20);
    name.setLocation(100, 100);
    c.add(name);

    tname = new JTextField();
    tname.setFont(new Font("Arial", Font.PLAIN, 15));
    tname.setSize(190, 20);
    tname.setLocation(200, 100);
    c.add(tname);

    mno = new JLabel("Mobile");
    mno.setFont(new Font("Arial", Font.PLAIN, 20));
    mno.setSize(100, 20);
    mno.setLocation(100, 150);
    c.add(mno);

    tmno = new JTextField();
    tmno.setFont(new Font("Arial", Font.PLAIN, 15));
    tmno.setSize(150, 20);
    tmno.setLocation(200, 150);
```

```
c.add(tmno);

gender = new JLabel("Gender");
gender.setFont(new Font("Arial", Font.PLAIN, 20));
gender.setSize(100, 20);
gender.setLocation(100, 200);
c.add(gender);

male = new JRadioButton("Male");
male.setFont(new Font("Arial", Font.PLAIN, 15));
male.setSelected(true);
male.setSize(75, 20);
male.setLocation(200, 200);
c.add(male);

female = new JRadioButton("Female");
female.setFont(new Font("Arial", Font.PLAIN, 15));
female.setSelected(false);
female.setSize(80, 20);
female.setLocation(275, 200);
c.add(female);

gengp = new ButtonGroup();
gengp.add(male);
gengp.add(female);

dob = new JLabel("DOB");
dob.setFont(new Font("Arial", Font.PLAIN, 20));
dob.setSize(100, 20);
dob.setLocation(100, 250);
c.add(dob);

date = new JComboBox(dates);
date.setFont(new Font("Arial", Font.PLAIN, 15));
date.setSize(50, 20);
date.setLocation(200, 250);
c.add(date);

month = new JComboBox(months);
month.setFont(new Font("Arial", Font.PLAIN, 15));
month.setSize(60, 20);
month.setLocation(250, 250);
c.add(month);

year = new JComboBox(years);
year.setFont(new Font("Arial", Font.PLAIN, 15));
year.setSize(60, 20);
year.setLocation(320, 250);
```

```
c.add(year);

add = new JLabel("Address");
add.setFont(new Font("Arial", Font.PLAIN, 20));
add.setSize(100, 20);
add.setLocation(100, 300);
c.add(add);

tadd = new JTextArea();
tadd.setFont(new Font("Arial", Font.PLAIN, 15));
tadd.setSize(200, 75);
tadd.setLocation(200, 300);
tadd.setLineWrap(true);
c.add(tadd);

term = new JCheckBox("Accept Terms And Conditions.");
term.setFont(new Font("Arial", Font.PLAIN, 15));
term.setSize(250, 20);
term.setLocation(150, 400);
c.add(term);

sub = new JButton("Submit");
sub.setFont(new Font("Arial", Font.PLAIN, 15));
sub.setSize(100, 20);
sub.setLocation(150, 450);
sub.addActionListener(this);
c.add(sub);

reset = new JButton("Reset");
reset.setFont(new Font("Arial", Font.PLAIN, 15));
reset.setSize(100, 20);
reset.setLocation(270, 450);
reset.addActionListener(this);
c.add(reset);

// Setting up the text
tout = new JTextArea();
tout.setFont(new Font("Arial", Font.ITALIC, 32));
tout.setSize(300, 400);
tout.setLocation(500, 100);
tout.setLineWrap(true);
tout.setEditable(false);
c.add(tout);

res = new JLabel("");
res.setFont(new Font("Arial", Font.PLAIN, 20));
res.setSize(500, 25);
res.setLocation(100, 500);
```

```
        c.add(res);

        resadd = new JTextArea();
        resadd.setFont(new Font("Arial", Font.PLAIN, 15));
        resadd.setSize(200, 75);
        resadd.setLocation(580, 175);
        resadd.setLineWrap(true);
        c.add(resadd);

        setVisible(true);
    }

    // method actionPerformed()
    // to get the action performed
    // by the user and act accordingly
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == sub) {
            if (term.isSelected()) {
                String data1;
                String data = "Name : " + tname.getText() + "\n" + "Mobile : " + tmno.getText() + "\n";
                if (male.isSelected())
                    data1 = "Gender : Male" + "\n";
                else
                    data1 = "Gender : Female" + "\n";
                String data2 = "DOB : " + (String) date.getSelectedItem() + "/" + (String) month.getSelectedItem() + "/"
                        + (String) year.getSelectedItem() + "\n";

                String data3 = "Address : " + tadd.getText();
                tout.setText(data + data1 + data2 + data3);
                tout.setEditable(false);
                res.setText("Registration Successfully..");
            } else {
                tout.setText("");
                resadd.setText("");
                res.setText("Please accept the" + " terms & conditions..");
            }
        }

        else if (e.getSource() == reset) {
            String def = "";
            tname.setText(def);
            tadd.setText(def);
            tmno.setText(def);
            res.setText(def);
            tout.setText(def);
            term.setSelected(false);
            date.setSelectedIndex(0);
            month.setSelectedIndex(0);
            year.setSelectedIndex(0);
            resadd.setText(def);
        }
    }
}
// Driver Code
class Registration {

    public static void main(String[] args) throws Exception {
        MyFrame f = new MyFrame();
    }
}
```

Output:



Registration Form

Name Aditya Ajmera

Mobile 911

Gender ● Male ○ Female

DOB 6 Apr 2001

Address 101, Khandwala Building

☑ Accept Terms And Conditions.

Registration Successfully..

```
Name : Aditya Ajmer
a
Mobile : 911
Gender : Male
DOB : 6/Apr/2001
Address : 101, Khan
dwala Building
```

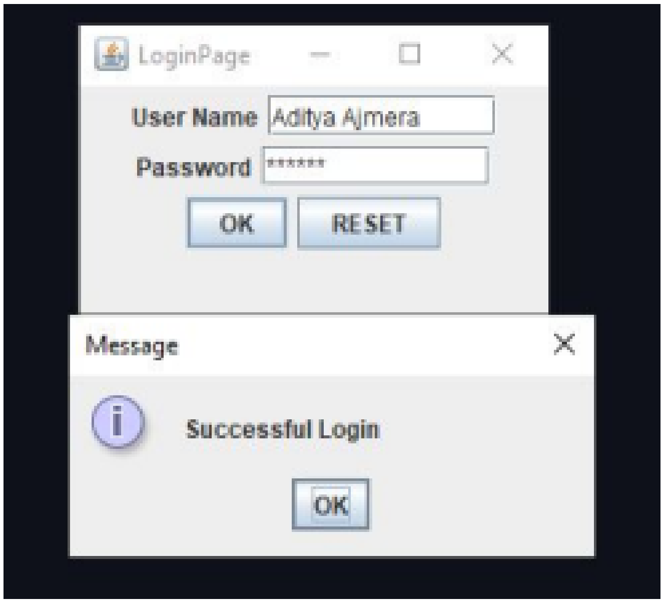
(ii) On Applet: Take a Login and Password from the user and display it on the third Text Field which appears only on clicking OK button and clear both the Text Fields on clicking RESET button

Code:

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

 import java.awt.event.*;

class Login extends JFrame implements ActionListener

{

Container c;

JLabel lblUserName,lblPassWord;

JTextField txtUserName;

JPasswordField txtPassWord;

JButton btnOK, btnRESET;

String strUserName, strPassWord;


Login()

{

c=getContentPane();

c.setLayout(new FlowLayout());


lblUserName=new JLabel("User

Name"); lblPassWord=new

JLabel("Password");

txtUserName=new JTextField(10);

txtPassWord=new JPasswordField(10);

txtPassWord.setEchoChar('*');

btnOK=new JButton("OK");

btnRESET=new JButton("RESET");
```

```
c.add(lblUserName);
c.add(txtUserName);
c.add(lblPassWord);
c.add(txtPassWord);
c.add(btnOK);
c.add(btnRESET);



btnOK.addActionListener(this);
btnRESET.addActionListener(this);


}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==btnOK)
{
strUserName=txtUserName.getText();
strPassWord=txtPassWord.getText();
if(strUserName.equals("Aditya Ajmera") && strPassWord.equals("qwerty"))
{
JOptionPane.showMessageDialog(c,"Successful Login");
}
else
{
JOptionPane.showMessageDialog(c,"Unsuccessful Login");
txtUserName.requestFocus();

}

}

else if(ae.getSource()==btnRESET)

{

txtUserName.setText("");

txtPassWord.setText("");

txtUserName.requestFocus();

}

else

{

System.exit(0);

}

}



public static void main(String z[]){

Login frm=new Login();

frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

frm.setBounds(500,500,250,250);

frm.setVisible(true);

frm.setTitle("LoginPage");

}

}
```

Output:



User Name Aditya Ajmera

Password ******

Successful Login

Experiment 16

Aim:

Develop simple swing applications and complex GUI using Java Swing classes.

(i) Write a program to create a window with four text fields for the name, street, city and pin code with suitable labels. Also windows contains a button MyInfo. When the user types the name, his street, city and pincode and then clicks the button, the types details must appear in Arial Font with Size 32, Italics.

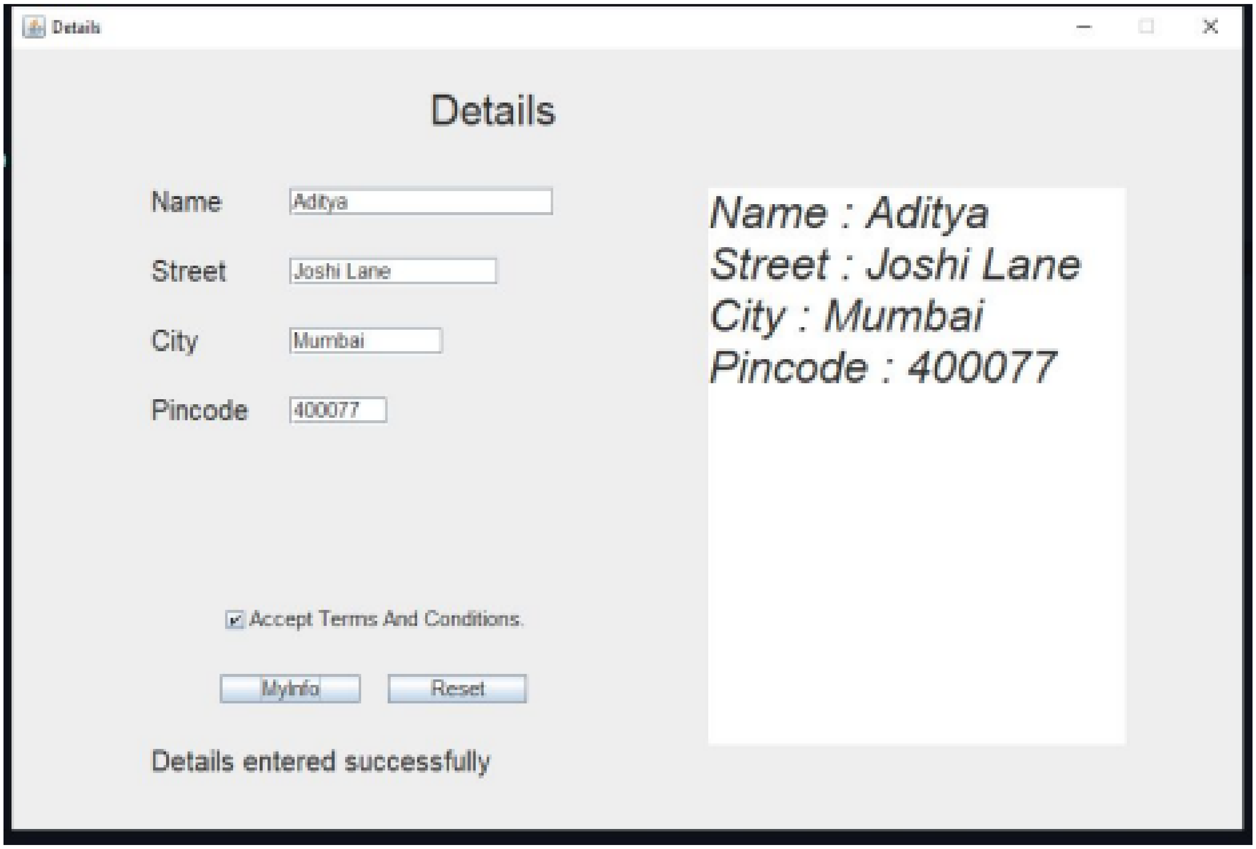
Code:

```
package experiment_16;

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

class MyFrame extends JFrame implements ActionListener {

    private Container c;
    private JLabel title;
    private JLabel name;
    private JTextField tname;
    private JLabel street;
    private JTextField tstreet;
    private JLabel city;
    private JTextField tcity;
    private JLabel pincode;
    private JTextField tpincode;
    private JCheckBox term;
    private JButton myinfo;
    private JButton reset;
    private JTextArea tout;
    private JLabel res;
    private JTextArea resadd;

    public MyFrame() {
        setTitle("Details");
        setBounds(300, 90, 900, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setResizable(false);
        c = getContentPane();
        c.setLayout(null);
        title = new JLabel("Details");
        title.setFont(new Font("Arial", Font.PLAIN, 30));
        title.setSize(300, 30);
        title.setLocation(300, 30);
        c.add(title);

        name = new JLabel("Name");
        name.setFont(new Font("Arial", Font.PLAIN, 20));
        name.setSize(100, 20);
        name.setLocation(100, 100);
        c.add(name);
        tname = new JTextField();
        tname.setFont(new Font("Arial", Font.PLAIN, 15));
        tname.setSize(190, 20);
        tname.setLocation(200, 100);
        c.add(tname);
        street = new JLabel("Street");
        street.setFont(new Font("Arial", Font.PLAIN, 20));
        street.setSize(100, 20);
        street.setLocation(100, 150);
        c.add(street);
        tstreet = new JTextField();
        tstreet.setFont(new Font("Arial", Font.PLAIN, 15));
        tstreet.setSize(150, 20);
        tstreet.setLocation(200, 150);
        c.add(tstreet);
        city = new JLabel("City");
```

```
city.setFont(new Font("Arial", Font.PLAIN, 20));
city.setSize(100, 20);
city.setLocation(100, 200);
c.add(city);
tcity = new JTextField();
tcity.setFont(new Font("Arial", Font.PLAIN, 15));
tcity.setSize(110, 20);
tcity.setLocation(200, 200);
c.add(tcity);
pincode = new JLabel("Pincode");
pincode.setFont(new Font("Arial", Font.PLAIN, 20));
pincode.setSize(100, 20);
pincode.setLocation(100, 250);
c.add(pincode);
tpincode = new JTextField();
tpincode.setFont(new Font("Arial", Font.PLAIN, 15));
tpincode.setSize(70, 20);
tpincode.setLocation(200, 250);
c.add(tpincode);
term = new JCheckBox("Accept Terms And Conditions.");
term.setFont(new Font("Arial", Font.PLAIN, 15));
term.setSize(250, 20);
term.setLocation(150, 400);
c.add(term);

myinfo = new JButton("MyInfo");
myinfo.setFont(new Font("Arial", Font.PLAIN, 15));
myinfo.setSize(100, 20);
myinfo.setLocation(150, 450);
myinfo.addActionListener(this);
c.add(myinfo);
reset = new JButton("Reset");
reset.setFont(new Font("Arial", Font.PLAIN, 15));
reset.setSize(100, 20);
reset.setLocation(270, 450);
reset.addActionListener(this);
c.add(reset);
tout = new JTextArea();
tout.setFont(new Font("Arial", Font.ITALIC, 32));
tout.setSize(300, 400);
tout.setLocation(500, 100);
tout.setLineWrap(true);
tout.setEditable(false);
c.add(tout);
res = new JLabel("");
res.setFont(new Font("Arial", Font.PLAIN, 20));
res.setSize(500, 25);
```

```
        res.setLocation(100, 500);
        c.add(res);

        resadd = new JTextArea();
        resadd.setFont(new Font("Arial", Font.PLAIN, 15));
        resadd.setSize(200, 75);
        resadd.setLocation(580, 175);
        resadd.setLineWrap(true);
        c.add(resadd);

        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == myinfo) {
            if (term.isSelected()) {
                String data1;
                String data = "Name : " + tname.getText() + "\n" + "Street : " + tstreet.getText() + "\n" + "City : "
                        + tcity.getText() + "\n" + "Pincode : " + tpincode.getText() + "\n";
                tout.setText(data);
                tout.setEditable(false);
                res.setText("Details entered successfully");
            } else {
                tout.setText("");
                resadd.setText("");
                res.setText("Please accept the" + " terms & conditions..");
            }
        } else if (e.getSource() == reset) {
            String def = "";
            tname.setText(def);
            tstreet.setText(def);
            tcity.setText(def);
            tpincode.setText(def);
            res.setText(def);
            tout.setText(def);
            term.setSelected(false);
            resadd.setText(def);

        }
    }
}

class Details {

    public static void main(String[] args) throws Exception {
        MyFrame f = new MyFrame();
    }
}
```

Output:



Name Aditya

Street Joshi Lane

City Mumbai

Pincode 400077

☑ Accept Terms And Conditions.

Details entered successfully

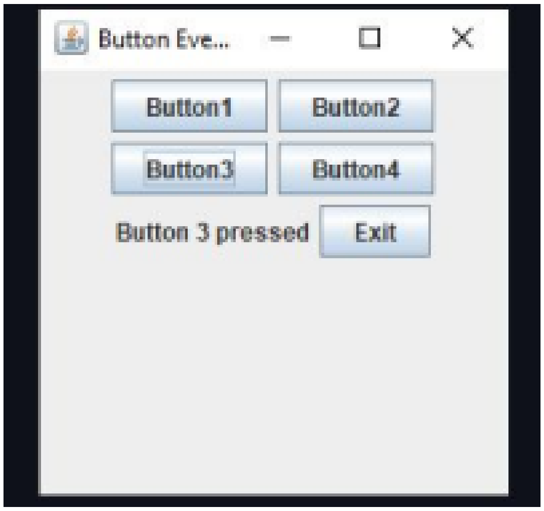
```
Name : Aditya
Street : Joshi Lane
City : Mumbai
Pincode : 400077
```

(ii) WA applet with 4 swing buttons with suitable texts on them. When the user presses a button a message should appear in the label as to which butt
on was pressed by the user

Code:

```
package experiment_16;

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

class btnPress extends JFrame implements ActionListener {
   Container c;
   JLabel L1;
   JButton B1, B2, B3, B4, exit;

   btnPress() {
        c = getContentPane();
        c.setLayout(new FlowLayout());

        B1 = new JButton("Button1");
        B2 = new JButton("Button2");
        B3 = new JButton("Button3");
        B4 = new JButton("Button4");
        L1 = new JLabel(" ");
        exit = new JButton("Exit");

        c.add(B1);
        c.add(B2);
        c.add(B3);
        c.add(B4);
        c.add(L1);
        c.add(exit);

        B1.addActionListener(this);
        B2.addActionListener(this);
        B3.addActionListener(this);
        B4.addActionListener(this);
        exit.addActionListener(this);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == B1) {
            L1.setText("Button 1 pressed");
        } else if (ae.getSource() == B2) {
            L1.setText("Button 2 pressed");
        } else if (ae.getSource() == B3) {
            L1.setText("Button 3 pressed");
        } else if (ae.getSource() == B4) {
            L1.setText("Button 4 pressed");
        } else if (ae.getSource() == exit) {
            System.exit(0);
        } else {
        }
    }

    public static void main(String z[]) {
        btnPress frm = new btnPress();
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frm.setBounds(200, 200, 250, 250);
        frm.setVisible(true);
        frm.setTitle("Button Event");
    }
}
```

Output:



Button 3 pressed