Experiment 2

Working with Conditional, Iteration and Jump Statements

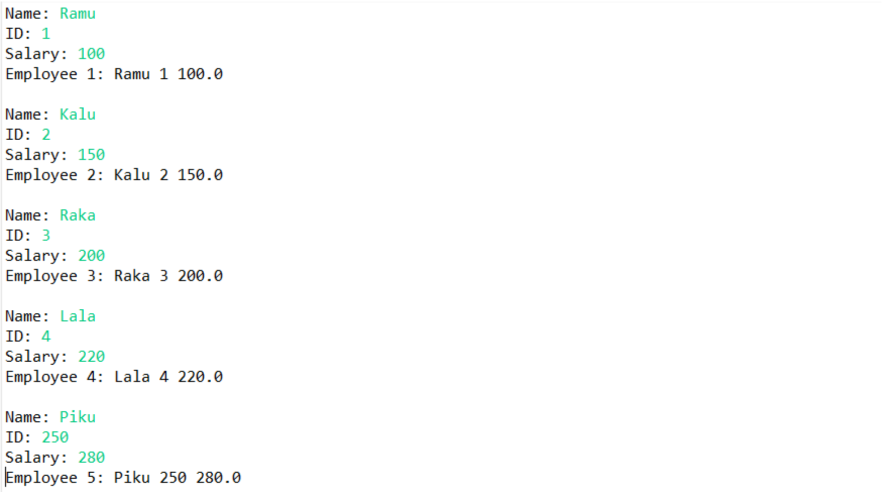
Aim of the experiment- To comprehend and apply conditional, iteration, and jump statements in programming to control program flow and execute specific actions based on conditions or loops.

Problem2.1: Wap to display the employee details of 5 employees using scanner class.

Solution2.1:

```
package experiment2;

import java.util.Scanner;

public class EmployeeDetails {

   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      for (int i = 1; i <= 5; i++) {

         System.out.print("Name: ");

         String name = sc.next();

         System.out.print("ID: ");

         int id = sc.nextInt();

         System.out.print("Salary: ");

         double salary = sc.nextDouble();

         System.out.println("Employee " + i + ": " + name + " " + id + " " + salary);

         System.out.println();

      }

      sc.close();

   }

}
```

Output2.1:



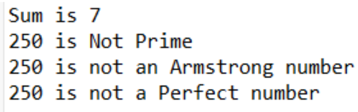
```
Name: Ramu
ID: 1
Salary: 100
Employee 1: Ramu 1 100.0

Name: Kalu
ID: 2
Salary: 150
Employee 2: Kalu 2 150.0

Name: Raka
ID: 3
Salary: 200
Employee 3: Raka 3 200.0

Name: Lala
ID: 4
Salary: 220
Employee 4: Lala 4 220.0

Name: Piku
ID: 250
Salary: 280
Employee 5: Piku 250 280.0
```

Problem2.2: Wap to check if the sum of all the digits of an input number is prime, armstrong, perfect or not.

Solution2.2:

```
package experiment2;
public class CheckNo {
        public static void main(String[] arg) {
                int num = 250, sum = 0;
                int temp = num;
                while(temp > 0) {
                        sum = sum + temp % 10;
                        temp = temp/10;
                }
                System.out.print("Sum is "+ sum + "\n");
                if(num % 2 ==0 || num % 3 ==0 || num % 5 == 0 || num % 7 == 0) {
                        System.out.print(num + " is Not Prime\n");
                }
                else {
                        System.out.print(num + " is Prime\n");
                }
                temp = num;
                while (temp > 0) {
       int digit = temp % 10;
       sum = sum + (digit * digit * digit);
       temp = temp / 10;
    }
    if (sum == num) {
       System.out.println(num + " is an Armstrong number");
    } else {
       System.out.println(num + " is not an Armstrong number");
    }
    int sumPerfect = 0;
    for (int i = 1; i <= num / 2; i++) {
       if (num % i == 0) {
         sumPerfect += i;
       }
    }
    if (sumPerfect == num) {
       System.out.println(num + " is a Perfect number");
    } else {
            System.out.println(num + " is not a Perfect number");
        }
               }
}
```

Output2.2:



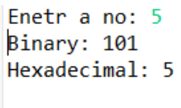
```
Sum is 7
250 is Not Prime
250 is not an Armstrong number
250 is not a Perfect number
```

Problem2.3: Wap to convert an input number into its respective binary and hexadecimal(without using any builtin library or method)

Solution2.3:

```
package experiment2;
import java.util.Scanner;
public class ConvertBase {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enetr a no: ");
        int num = sc.nextInt();
        int temp = num;
        String binary = "";
        while (temp > 0) {
            binary = (temp % 2) + binary;
            temp /= 2;
        }
        System.out.println("Binary: " + binary);
        temp = num;
        String hex = "";
        char[] h = {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
        while (temp > 0) {
            hex = h[temp % 16] + hex;
            temp /= 16;
        }
        System.out.println("Hexadecimal: " + hex);
        sc.close();
    }
}
```

Output2.3:



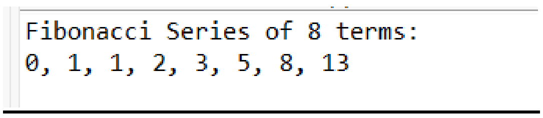
```
Enetr a no: 5
Binary: 101
Hexadecimal: 5
```

Problem2.4 Wap to implement both recursive and non-recursive functions to find the fibonacci series.

Solution2.4:

```
package experiment2;

public class Fiborec {

           public static int fibonacci(int n) {

      if (n <= 1) {

         return n;

      }

      return fibonacci(n - 1) + fibonacci(n - 2);

   }

   public static void main(String[] args) {

      int terms = 8;

      System.out.println("Fibonacci Series of " + terms + " terms:");

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

         System.out.print(fibonacci(i));

         if (i < terms - 1) {

            System.out.print(", ");

         }}}}

package experiment2;

public class FiboNoRec {

              public static void main(String[] args) {

                 int n = 8;

                 int firstTerm = 0;

                 int secondTerm = 1;

                 System.out.println("Fibonacci Series till " + n + " terms:");

                 for (int i = 1; i <= n; ++i) {

                    System.out.print(firstTerm + ", ");
                    int nextTerm = firstTerm + secondTerm;

                    firstTerm = secondTerm;

                    secondTerm = nextTerm;

                 }}}
```

Output2.4:



```
Fibonacci Series of 8 terms:
0, 1, 1, 2, 3, 5, 8, 13
```

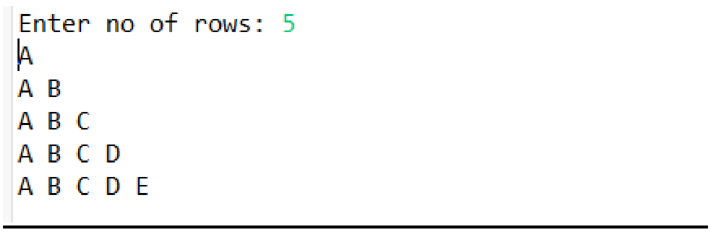
Problem2.5.1: Wap to print the following pattern (Number of columns/levels must be taken input by the user).
(PATTERN 1)

A
A B
A B C
A B C D
A B C D E

Solution2.5.1:

```
package experiment2;

import java.util.Scanner;

public class Pattern1 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

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

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

                System.out.print((char)('A' + j) + " ");

            System.out.println();

        }

        sc.close();

    }

}
```

Output2.5.1:



```
Enter no of rows: 5
A
A B
A B C
A B C D
A B C D E
```

Problem2.5.2: Wap to print the following pattern (Number of columns/levels must be taken input by the user).

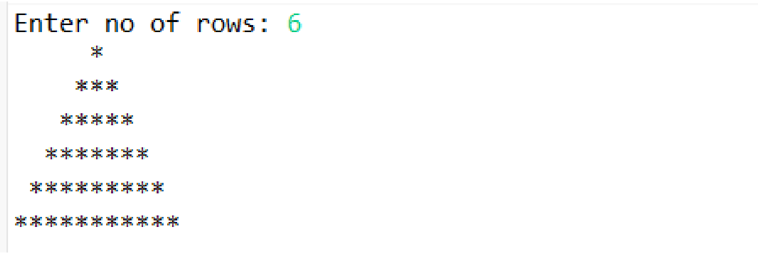
(PATTERN 2)

```
        *
      *    *
    *    *    *
  *   *    *    *
 *  *    *    *   *
*  *  *    *    *    *
```

Solution2.5.2:

```
package experiment2;
import java.util.Scanner;
public class Pattern2 {
            public static void main(String[] args) {
       Scanner sc = new Scanner(System.in);
       int n = sc.nextInt();
       for (int i = 1; i <= n; i++) {
          for (int s = 1; s <= n - i; s++)
             System.out.print(" ");
          for (int j = 1; j <= (2 * i - 1); j++)
             System.out.print("*");
          System.out.println();
       }
       sc.close();
    }
}
```

Output2.5.2:



```
Enter no of rows: 6
     *
    ***
   *****
  *******
 *********
***********
```

--Submitted By--
Name-Arjun Ojha
Regd_no-2401020055
Branch-CSE
Sem-4th
Gr-01