

An internship report on **JAVA PROGRAMMING**

*A report submitted in partial fulfilment of the requirements for the curriculum of 6th
Semester, Degree of*

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

By
SACHIN KUMAR
Regd. No: 20033445030

(Duration- 27th March 2023 - 23rd April 2023)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAMGARH ENGINEERING COLLEGE

(ESTD. By Govt. Of Jharkhand & Run by Techno India)

Approved by AICTE, New Delhi and affiliated to JUT, Ranchi

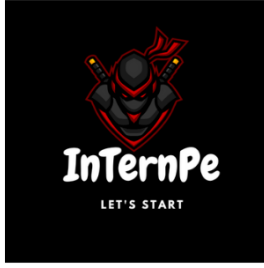
ISO 9001:2015, Certified Institute

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RAMGARH ENGINEERING COLLEGE
(ESTD. By Govt. Of Jharkhand & Run by Techno India)



This is to certify that the one month virtual “**Internship report**” submitted by **SACHIN KUMAR** (Regd. No. 20033445030) in the domain “**JAVA PROGRAMMING**” (27th March to 23rd April) is work done by him and submitted during 2023-2024 academic year, in partial fulfillment of the requirements for the curriculum of 6th Semester, degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING**, at **INTERN PE**.

Prof. Ashim Kumar Mahto
Head Of the Department
Department Of CSE



@internPE

COMPLETION CERTIFICATE

CID : IPI#3281

To whomever it may concern

This is to certify that **Sachin Kumar** worked as a Intern in our company from **27-Mar-2023 to 23-Apr-2023**

Please find the internship details below:

Company Name: InternPe

Location: Remote

Domain: JAVA Programming

Designation: Intern

During their working period, we found him/her to be a sincere and dedicated intern with a professional attitude and very good knowledge of the job.

We thank him/her for their efforts and contribution and wish him/her the best in future endeavors.

Yours Sincerely

(Co-Founder)
InternPe



Table of Contents

1. Acknowledgements

2. Objectives

3. Introduction

4. Tasks

A. Assignments

- i) Create an infinite loop from scratch
- ii) Create an executable comment
- iii) Create a named loop
- iv) Find a single duplicate in an array
- v) Find a non-unique duplicate of an array an integer

B. Rock Paper Scissor Game

- i) Code
- ii) Code Walkthrough
- iii) Conclusion

5. About Internship

6. Conclusion

Acknowledgements

I would like to express our deepest appreciation to all those who provided me the possibility to complete this Internship. A special gratitude to the **Team of InternPe** for giving me the opportunity to do an internship within the organization. Your patience and understanding allowed me to grow and learn from my experiences, making this internship an enriching and rewarding one.

A special thanks goes to the Head of the Department, **Prof. Ashim Kumar Mahto** whose suggestions and guidance helped me and specially encouraged me for making this report, I also appreciate the guidance and help received from all other teachers.

I am extremely grateful to my friends and the fellow interns who helped me in successful completion of this internship.

Sachin Kumar
(20033445030)

Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
- Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.
- This internship is a task-based internship the so, focus is on accomplishing specific objectives or assignments and need not follow the traditional approach.

Introduction

Java is a versatile, high-level, object-oriented programming language that has gained immense popularity since its inception in 1995. Developed by James Gosling and his team at Sun Microsystems (now owned by Oracle Corporation), Java was designed to be platform-independent, enabling developers to write code that could run on any device or operating system. Over the years, Java has become a cornerstone of the software development industry, powering a wide range of applications from web and mobile to enterprise-level systems.

Key Features of Java:

Platform Independence: Java's "Write Once, Run Anywhere" (WORA) principle allows developers to create applications that can be executed on different platforms without modification, thanks to the Java Virtual Machine (JVM) that acts as an intermediary between the code and the underlying hardware.

Object-Oriented Programming (OOP): Java follows the OOP paradigm, enabling developers to model real-world entities using classes and objects. This approach promotes code reusability, maintainability, and scalability.

Garbage Collection: Java features an automatic garbage collector that manages memory allocation and frees up resources when objects are no longer in use. This reduces the burden on developers for memory management.

Exception Handling: Java provides a robust mechanism for handling errors and exceptions, making code more robust and reliable.

Rich Standard Library: Java comes with an extensive standard library (Java API), offering a wide range of pre-built classes and methods for various tasks, such as networking, file I/O, and data structures.

Applications of Java:

Web Development: Java has been widely used for creating web applications through technologies like JavaServer Pages (JSP), Servlets, and the Spring Framework.

Android App Development: Java has been a primary programming language for developing Android mobile applications.

Enterprise Applications: Java's scalability and reliability make it a popular choice for building large-scale enterprise applications and systems.

Internet of Things (IoT): Java's lightweight version, Java ME (Micro Edition), is used in developing applications for IoT devices.

Big Data and Analytics: Java is employed in various big data processing frameworks like Apache Hadoop and Apache Spark.

Tasks

As I have enrolled for a task-based internship which typically refers to an internship program where interns are assigned specific tasks or projects to complete during their internship period. Instead of following a traditional time-based approach where interns work for a fixed number of hours each day, the focus is on accomplishing specific objectives or assignments.

In a task-based internship, interns may work on real projects that align with the company's goals and provide meaningful learning experiences. The tasks given to interns are usually relevant to their field of study or career interests, allowing them to gain practical skills and hands-on experience.

Here are some characteristics of task-based internships:

- **Goal-Oriented:** The internship revolves around achieving specific goals or completing designated tasks rather than merely fulfilling a set number of hours.
- **Project-Centered:** Interns may be assigned to work on individual projects or collaborate with teams on ongoing initiatives.
- **Skill Development:** The tasks assigned aim to help interns develop specific skills related to their chosen career path.
- **Practical Experience:** Interns get the opportunity to apply theoretical knowledge to real-world scenarios, enhancing their understanding and expertise.

- **Evaluation:** Interns may be assessed based on their performance and the quality of work completed during the internship.
- **Flexibility:** Task-based internships may offer more flexibility in terms of work hours, as long as the assigned tasks are completed within the specified timeframe.

Tasks completed by me:

To fulfill the eligibility criteria, I have completed some of the tasks using Java programming:

1. Assignments

i) Create an infinite loop from scratch

```
Code:- class Infinte_loop{
    public static void main(String[] args) {
        do{
            System.out.println("This is infinte loop");
        } while(true);
    }
}
```

Output:-

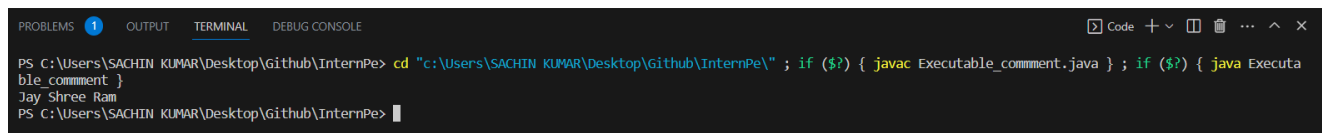


```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
This is infinte loop
This is infinte loop
This is infinte loop
This is infinte loop
This is infinte loop
This is infinte loop
This is infinte loop
This is infinte loop
This is infinte loop
```

To be continued...

ii) Create an executable comment

```
public class Executable_commmment {
    public static void main(String[] args) {
        ///--- The comment below is executable comment ---//
        //\u000d System.out.println("Jay Shree Ram");
    }} Output:-
```



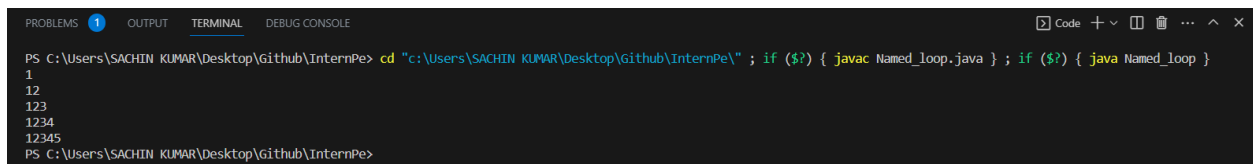
```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe> cd "c:\Users\SACHIN KUMAR\Desktop\Github\InternPe\" ; if ($?) { javac Executable_comment.java } ; if ($?) { java Executable_commmment }
Jay Shree Ram
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe>
```

iii) Create a named loop

Code:-

```
class Named_loop {
    public static void main(String args[]) {
        outer:
        //label
        for (int i = 1;i <= 5;i++) {
            inner:
            //label
            for(int j = 1;j<=i; j++) {
                System.out.print(j+"");
                if (j == 5)
                    break inner;
            }
            System.out.println("");
            continue outer;
        }
    }
}
```

Output:-



```
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe> cd "C:\Users\SACHIN KUMAR\Desktop\Github\InternPe\" ; if ($?) { javac Named_loop.java } ; if ($?) { java Named_loop }
1
12
123
1234
12345
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe>
```

iv) Find a single duplicate in an array

Code:-

```
import java.util.ArrayList;

public class Single_duplicate_array {
    static void findDuplicates(
        int arr[], int len)
    {
        // initialize ifPresent as false
        boolean ifPresent = false;
        // ArrayList to store the output
        ArrayList<Integer> al = new ArrayList<Integer>();

        for (int i = 0; i < len - 1; i++) {
            for (int j = i + 1; j < len; j++) {
                if (arr[i] == arr[j]) {
```

```

        // checking if element is
        // present in the ArrayList
        // or not if present then break
        if (al.contains(arr[i])) {
            break;
        }
        else {
            al.add(arr[i]);
            ifPresent = true;
        }
    }
}
}
if (ifPresent == true) {

    System.out.print(al + " ");
}
else {
    System.out.print(
        "No duplicates present in arrays");
}
}
// Driver Code
public static void main(String[] args)
{

    int arr[] = { 1, 1, 2, 3, 5, 6, 5, 4, 7 };
    int n = arr.length;

    findDuplicates(arr, n);
}

```

Output:-

```

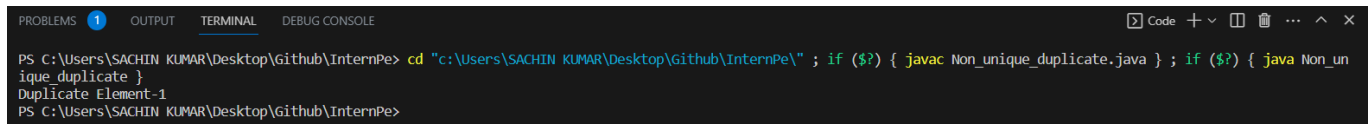
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe> cd "c:\Users\SACHIN KUMAR\Desktop\Github\InternPe\" ; if ($?) { javac Single_duplicate_array.java } ; if ($?) { java Sing
le_duplicate_array }
[1, 5]
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe>

```

v) Find a non-unique duplicate of an array an integer

```
Code:- import java.util.HashSet;
public class Non_unique_duplicate {
    public static int Task5NonDuplicate(int[]nums) {
        HashSet<Integer> set = new HashSet<>();
        for (int num : nums) {
            if(set.contains(num)){
                return num;
            }
            set.add(num);
        }
        return -1;//no duplicate Found
    }
    public static void main(String args[]){
        int[]nums={1,2,3,4,5};
        int duplicate=Task5NonDuplicate(nums);
        System.out.println("Duplicate Element" +duplicate);
    }
}
```

Output:-



```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe> cd "c:\Users\SACHIN KUMAR\Desktop\Github\InternPe\" ; if ($?) { javac Non_unique_duplicate.java } ; if ($?) { java Non_un
ique_duplicate }
Duplicate Element-1
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe>
```

2. Rock Paper Scissor Game

A.Rock Paper Scissor Game:

This Java program implements a rock paper scissor game. The game randomly generates a move among rock, paper and scissors. The logic of rock paper scissors is a simple one. Each of the three choices (rock, paper, scissors) beats one of the other two, and loses to the other.

Rock beats scissors (rock crushes scissors)

Paper beats rock (paper covers rock)

Scissors beats paper (scissors cut paper)

If both players choose the same thing, it is a tie.

This creates a 3-way win-lose-tie system. There is no way to always win, but there are strategies that can be used to increase your chances of winning.

Code:-

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

public class RockPaperScissors {

    public static void main(String[] args) {
        // Create a scanner object for user input
        Scanner scanner = new Scanner(System.in);

        // Create a random object for computer choice
        Random random = new Random();

        // Create an array of moves
        String[] moves = {"rock", "paper", "scissors"};

        // Create variables to store the scores
        int userScore = 0;
        int computerScore = 0;

        // Create a loop for 10 rounds
        for (int i = 0; i < 10; i++) {
            // Prompt the user to enter their move
            System.out.println("Round " + (i + 1) + ": Enter your move (rock, paper, or scissors):");
            String userMove = scanner.nextLine().toLowerCase();

            // Validate the user move
            if (!userMove.equals("rock") && !userMove.equals("paper") &&
!userMove.equals("scissors")) {
                System.out.println("Invalid move. Try again.");
                i--; // Repeat the round
                continue; // Skip the rest of the loop
            }

            // Generate a random move for the computer
```

```

int computerIndex = random.nextInt(3);
String computerMove = moves[computerIndex];

// Display the moves
System.out.println("You chose: " + userMove);
System.out.println("Computer chose: " + computerMove);

// Compare the moves and update the scores
if (userMove.equals(computerMove)) {
    System.out.println("It's a tie!");
} else if (userMove.equals("rock") && computerMove.equals("scissors") ||
    userMove.equals("paper") && computerMove.equals("rock") ||
    userMove.equals("scissors") && computerMove.equals("paper")) {
    System.out.println("You win!");
    userScore++;
} else {
    System.out.println("You lose!");
    computerScore++;
}

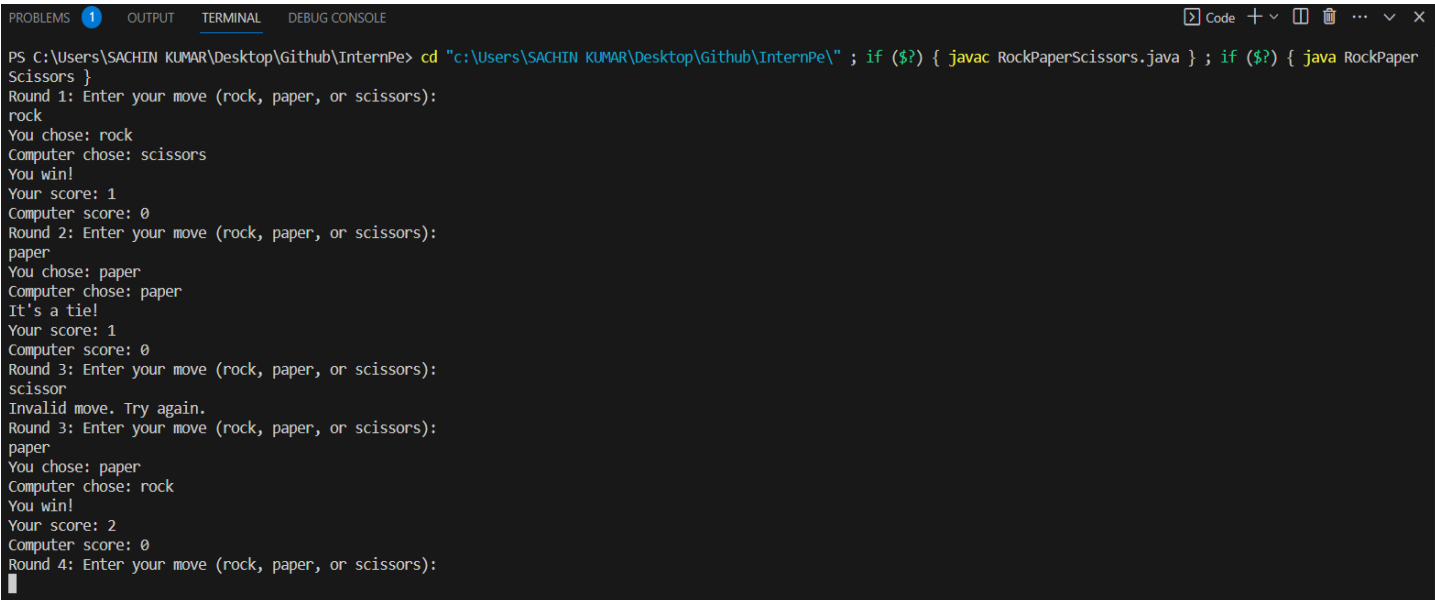
// Display the scores
System.out.println("Your score: " + userScore);
System.out.println("Computer score: " + computerScore);
}

// Display the final result
if (userScore > computerScore) {
    System.out.println("You won the game!");
} else if (userScore < computerScore) {
    System.out.println("You lost the game!");
} else {
    System.out.println("The game ended in a draw!");
}

// Close the scanner object
scanner.close();
}
}

```

Output:-



```
PS C:\Users\SACHIN KUMAR\Desktop\Github\InternPe> cd "c:\Users\SACHIN KUMAR\Desktop\Github\InternPe\" ; if ($?) { javac RockPaperScissors.java } ; if ($?) { java RockPaperScissors }
Round 1: Enter your move (rock, paper, or scissors):
rock
You chose: rock
Computer chose: scissors
You win!
Your score: 1
Computer score: 0
Round 2: Enter your move (rock, paper, or scissors):
paper
You chose: paper
Computer chose: paper
It's a tie!
Your score: 1
Computer score: 0
Round 3: Enter your move (rock, paper, or scissors):
scissor
Invalid move. Try again.
Round 3: Enter your move (rock, paper, or scissors):
paper
You chose: paper
Computer chose: rock
You win!
Your score: 2
Computer score: 0
Round 4: Enter your move (rock, paper, or scissors):
```

Code walkthrough

- First, we import the Scanner and Random classes from the java.util package. These classes will help us get the user input and generate random numbers.
- Next, we define the RockPaperScissors class and the main method. This is where our program starts.
- Inside the main method, we create a scanner object called scanner. This object will allow us to read the user input from the console.
- We also create an array of strings called moves. This array stores the three possible moves: rock, paper, and scissors.
- We then create two variables to store the scores of the user and the computer. We initialize them to zero.
- We then create a for loop that runs for 10 rounds. Each round consists of the following steps:
 - We prompt the user to enter their move by printing a message to the console. We use the scanner object to read the user input and store it in a string variable called userMove. We also convert the user input to lowercase to make it easier to compare.
 - We validate the user move by checking if it is equal to one of the elements in the moves array. If not, we print an error message and repeat the round by decrementing the loop counter and continuing to the next iteration.

- We generate a random move for the computer by creating a random object called `random` and calling its `nextInt` method with an argument of 3. This method returns a random integer between 0 and 2, which we use as an index to access an element from the `moves` array. We store this element in a string variable called `computerMove`.
- We display the moves of both players by printing them to the console.
- We compare the moves and update the scores by using if-else statements. We use the rules of rock paper scissors to determine who wins, loses, or ties. If the user wins, we increment their score by one and print a message. If the computer wins, we increment their score by one and print a message. If both players choose the same move, we print a message indicating a tie.
- We print a blank line for spacing between rounds.
- After the loop ends, we print the final scores of both players by using string concatenation and printing them to the console.
- We declare a winner based on the scores by using if-else statements. If the user score is greater than the computer score, we print a message saying that the user is the winner. If the computer score is greater than the user score, we print a message saying that the computer is the winner. If both scores are equal, we print a message saying that it is a draw.
- Finally, we close the scanner object by calling its `close` method.

About Internship

Some important points we have to remember during this internship tenure –

- Update your LinkedIn Profile and share all your achievements (Offer Letter / Internship Completion Certificate) which you got from us and tag Intern Pe, also use #InternPe.
- If your project/code is found copied, your internship will be terminated and you will be banned from further opportunities from us.
- Share video of the completed task on LinkedIn and tag Intern Pe also use #oasisinfobyte (Explanation is Mandatory).
- For Tech Internship maintain a separate GitHub repository (name as Intern Pe for all the tasks and share the link of the GitHub repository in the task submission form (it will be given later through email).

I successfully fulfilled each criteria and finally awarded with the Internship certificate.

One can easily apply for this virtual internship by visiting the official website “<https://internpe.in/>”. It is an unpaid and task-based internship which can be easily done from anywhere remotely.

Conclusion

This internship covers essential aspects of Java programming.

It is task-based internships, and demonstrates one basic practical Java programs.

The basic of Java program covers, syntax rule, variable and data types, expression and operators along with loops and conditional statement

The second is the program

RockPapperScissorGame, is a simple interactive game where the user simply E the one of this(Rock Papper Scissor) after that hit the enter button and the

Both programs demonstrate fundamental programming concepts, including conditional statements, loops, functions, and user input/output handling. While these implementations serve as educational examples, they can be further enhanced to include error handling, input validation, and additional features for real-world applications.

For interns, task-based internships offer a goal-oriented and skill-focused learning experience. Working on specific projects allows them to apply theoretical knowledge to practical scenarios, develop relevant skills, and gain hands-on experience in their chosen field. The internship's flexibility and project-centered approach provide valuable learning opportunities, helping interns build a strong foundation for their future careers.