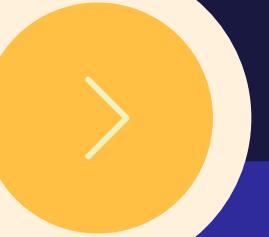




# TOPPER WORLD

## INTERNSHIP PROGRAM

JAVA  
PROGRAMMING

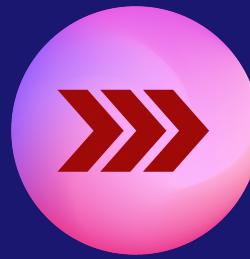


# Internship Roadmap

What You have To DO ?

## Step 1

LEARN



## Step 2

APPLY



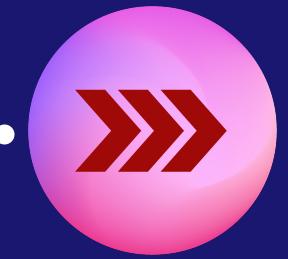
## Step 3

ENGAGE



## Step 4

SHARE



TOPPERWorld

# ABOUT US

- Topperworld is a project-based learning organization that aims to build a strong tech future for all developers.
- We at Topperworld strongly believe that practical knowledge can make a student more successful in their career.
- Our aim is to help others gain personal and professional skills for their careers.
- Topperworld is primarily for students who want to start a career in a technical field but have lack of basic knowledge.
- We are a officially MSME registered e-learning startup company.

# RULES AND CRITERIA

- To receive fast updates about internships, interns must follow us on social media.
- Selected interns are need to join WhatsApp group as well as Telegram group for Task Updates .
- All chosen interns must complete three task in order to be eligible for a Certificate of Completion.
- To be eligible for a Letter of Recommendation (LOR), complete all of the assigned tasks.
- If we discover that your code contains plagiarism, we will fire you by way of the internship.



# RULES AND CRITERIA

- Maintain a unique GitHub repository (for instance, TW Internship).
- Add all the task codes and projects to the GitHub repository.
- Upload the task videos with explanations to LinkedIn and tag @Topperworld.
- All interns must update their LinkedIn profiles.
- Change the title of your LinkedIn profile to reflect your position, such as "Java Programming intern at Topperworld."
- Update your LinkedIn Experience with Topperworld.



# JAVA PROGRAMMING



# TASK 1 : Number Guessing Game

## OBJECTIVE:

The objective of the number guessing game project in Java is to create an interactive and entertaining game where the player must guess a randomly generated number within a specified range. The game will provide feedback on each guess, guiding the player to find the correct number within a limited number of attempts.

## REQUIREMENTS:

- **Java Development Environment:** Ensure that you have JDK (Java Development Kit) installed on your system to write and compile Java code.
- **User Interface:** Implement a simple user interface to interact with the player. This can be done using the command-line interface (CLI) or a graphical user interface (GUI).
- **Random Number Generation:** Use Java's random number generator to generate a random number within a given range.

```
Welcome to Guess Number Game
You Will Be Asked To Guess A Number To Win The Game
You have Maximum 5 Attemp Limit
Enter a guess number between 1 to 100
75
Your Guess Number is Greater.
Enter a guess number between 1 to 100
45
Your Guess Number is Smaller.
Enter a guess number between 1 to 100
58
Your Guess Number is Smaller.
Enter a guess number between 1 to 100
67
Your Guess Number is Greater.
Enter a guess number between 1 to 100
62
OOOhOO!, Your Number is Correct. You Win the Game!
```

# **TASK 1 : Number Guessing Game**

- **User Input:** Allow the player to input their guesses using the user interface.
- **Feedback Mechanism:** Provide feedback to the player after each guess, indicating whether their guess is too high, too low, or correct.

## **STAGES:**

- **Project Setup:** Set up a new Java project using an Integrated Development Environment (IDE) or a text editor and compile your Java code.
- **Implement Random Number Generation:** Write code to generate a random number within the specified range. You can use the `java.util.Random` class for this.
- **Create User Interface:** Develop a simple user interface to display the game instructions and receive input from the player.
- **Accept User Input:** complement code to receive and validate the player's guesses from the user interface.

# TASK 1 : Number Guessing Game

---

- **Compare Guess with the Random Number:** Compare the player's guess with the randomly generated number and provide feedback accordingly (too high, too low, or correct).
- **Track Attempts:** Keep track of the number of attempts the player has made and limit the number of guesses allowed based on the game's design.
- **Implement Win/Lose Conditions:** Determine the conditions for winning or losing the game, and display appropriate messages to the player.
- **Optional:** Add Difficulty Levels (if desired): If you want to include different difficulty levels, create a mechanism to handle varying ranges and attempts allowed.
- **Test the Game:** Thoroughly test the game to identify and fix any bugs or issues.

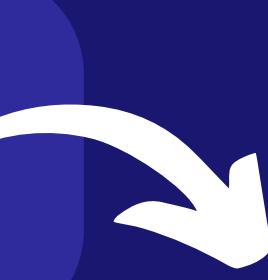
# TASK 2 : ATM Interface

## OBJECTIVE:

The objective of the ATM (Automated Teller Machine) interface project in Java is to create a simulation of an ATM system that allows users to perform various banking transactions, such as withdrawing cash, checking account balance, depositing money, and transferring funds between accounts. The project aims to provide a user-friendly and secure interface to interact with the ATM functionalities.

## REQUIREMENTS:

- **Java Development Environment:** Ensure that you have JDK (Java Development Kit) installed on your system to write and compile Java code.



```
C:\WINDOWS\system32\cmd.exe
Balance : 0
ATM Machine
Choose 1 for Withdraw
Choose 2 for Deposit
Choose 3 for Check Balance
Choose 4 for EXIT

Choose the operation:2
Enter money to be deposited:1000
Your Money has been successfully deposited

ATM Machine
Choose 1 for Withdraw
Choose 2 for Deposit
Choose 3 for Check Balance
Choose 4 for EXIT

Choose the operation:3
Balance : 1000
```

# TASK 2 : ATM Interface

- **User Interface:** Implement a graphical user interface (GUI) to interact with the user. The GUI should display options for different transactions and receive input from the user.
- **Account Management:** Design a system to manage user accounts, including storing account information, such as account number, PIN (Personal Identification Number), and account balance.
- **Transaction Processing:** Implement the logic to process various ATM transactions, such as cash withdrawals, balance inquiries, deposits, and fund transfers.

## STAGES:

- **Project Setup:** Set up a new Java project using an Integrated Development Environment (IDE) or a text editor and compile your Java code.
- **Create User Interface:** Develop a GUI for the ATM system that displays options for different transactions, such as withdrawing cash, checking balance, etc.

# TASK 2 : ATM Interface

---

- **User Authentication:** Implement a mechanism to prompt the user for their account number and PIN to authenticate them before allowing access to the ATM functionalities.
- **Account Management:** Design a data structure (e.g., arrays, lists, or database) to store and manage account information, including account numbers, PINs, and account balances.
- **Transaction Processing:** Write code to process different ATM transactions, such as cash withdrawals, balance inquiries, deposits, and fund transfers. Ensure that the transactions update the account balances accordingly.
- **Security Measures:** Implement code to validate the user's PIN and deny access to unauthorized users. Implement security measures to prevent unauthorized access to account information.

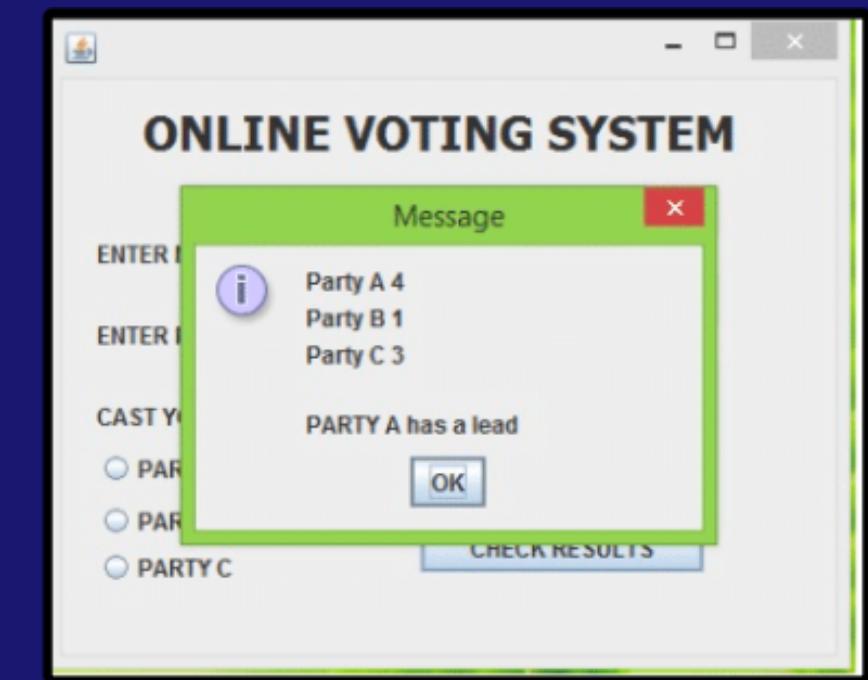
# TASK 3 : Online Voting System

## OBJECTIVE:

The objective of the online voting system project in Java programming is to develop a secure and efficient platform that allows users to cast their votes electronically for various elections or polls. The system aims to streamline the voting process, improve accessibility, and ensure the integrity of the voting results.

## REQUIREMENTS:

- **User Authentication:** Implement a secure user authentication system to verify the identity of voters and prevent unauthorized access.
- **Candidate Registration:** Allow candidates to register for elections and provide necessary information for voters to make informed choices.



# **TASK 3 : Online Voting System**

- **Ballot Creation:** Create electronic ballots for each election, listing the candidates and relevant voting options.
- **Vote Casting:** Enable users to cast their votes securely and privately for the candidates of their choice.
- **Vote Counting:** Implement a vote counting mechanism to tally the votes and determine the election results accurately.

## **STAGES:**

### **1.Design and Planning:**

- Plan the overall architecture of the online voting system, including the database schema and user interface design.
- Define the functionalities and features required for user authentication, candidate registration, voting, and result computation.

# **TASK 3 : Online Voting System**

## **2. User Authentication and Registration:**

- Implement a user registration system with proper validation and security measures.
- Set up user authentication using secure login credentials.

## **3. Candidate Registration and Ballot Creation:**

- Develop a candidate registration module where candidates can enter their details for each election.
- Create electronic ballots for each election, listing the registered candidates.

## **4. Vote Casting:**

- Implement the vote casting mechanism, ensuring that each user can cast only one vote per election.
- Ensure vote privacy and security during transmission and storage.

## **5. Vote Counting and Result Computation:**

- Implement a vote counting algorithm to tally the votes for each candidate in each election.
- Determine the election results based on the vote count.

# CONTACT US

If you have any doubts and queries feel free to contact us !

## CONNECT WITH US !



[topperworldinternship@gmail.com](mailto:topperworldinternship@gmail.com)



[www.topperworld.in](http://www.topperworld.in)



[Topperworld.in](https://www.instagram.com/topperworld.in)



[Topperworld](https://www.linkedin.com/company/topperworld/)



[Topperworld](https://t.me/Topperworld)



# THANK YOU

Stay Connected !

