

Assignment 1: Library Management System

Objective

Understand how to use final, static, and global variables/methods in a class-based context.

Instructions

1. Create a Java class named Library.
2. Define the following:
 - o A final variable MAX_BOOKS that represents the maximum number of books a library can hold. Set this to 1000.
 - o A static variable totalLibraries that keeps track of the number of Library instances created.
 - o A global (class-level) variable libraryName that specifies the name of the library.
 - o A final method printMaxBooks() that prints the maximum number of books.
 - o A static method incrementLibraryCount() that increments the totalLibraries variable.
 - o A non-static method displayLibraryName() that prints the name of the library.
3. In the Library class:
 - o Create a constructor that sets the library name and increments the total library count using the incrementLibraryCount() method.
 - o Override the toString() method to provide information about the library, including its name and maximum book capacity.
4. Create a Main class to test the Library class:
 - o Instantiate several Library objects.
 - o Call printMaxBooks() on each object.
 - o Call displayLibraryName() on each object.
 - o Print the total number of libraries using the totalLibraries variable.

sol:

```
public class Library {
    public final int MAX_BOOKS = 1000;
    public static int totalLibraries=0;
    String libraryName;

    final void printMaxBooks() {
        System.out.println("No.of Books: "+MAX_BOOKS);
    }
    static void incrementLibraryCount() {
        totalLibraries++;
    }
    void displayLibraryName() {
        System.out.println("Name of the library :"+libraryName);
    }

    public Library(String Libraryname) {
        this.libraryName=Libraryname;
        incrementLibraryCount();
    }
}
```

```

    }

    public String toString() {
        return "Library [MAX_BOOKS=" + MAX_BOOKS + ", libraryName=" +
libraryName + "]\n";
    }

    public static void main(String[] args) {
        Library l1 =new Library("NATIONAL LIBRARY");
        l1.printMaxBooks();
        //l1.incrementLibraryCount();
        l1.displayLibraryName();
        System.out.println();
        Library l2=new Library("PUBLIC LIBRARY");
        l2.printMaxBooks();
        //l2.incrementLibraryCount();
        l2.displayLibraryName();
        System.out.println();
        Library l3 =new Library("LIBRARY");
        l3.printMaxBooks();
        //l3.incrementLibraryCount();
        l3.displayLibraryName();
        System.out.println();
        System.out.println("Total number of Libraries
:"+totalLibraries);

    }
}

```

Assignment 2: Student Grades System

Objective

Practice the usage of final, static, and global variables and methods in a class for managing student grades.

Instructions

1. Create a Java class named Student.

2. Define the following:

- o A final variable MAX_GRADES representing the maximum number of grades a student can have. Set this to 10.

- o A static variable totalStudents to keep track of the number of Student objects created.

- o A global (class-level) variable schoolName that specifies the name of the school.

- o A final method printMaxGrades() that prints the maximum number of grades a student can have.

- o A static method incrementStudentCount() that increments the totalStudents variable.

- o A non-static method displaySchoolName() that prints the school name.

3. In the Student class:

- o Create a constructor that sets the student's name and increments the total student

count using the incrementStudentCount() method.
o Override the toString() method to provide information about the student, including their name and the maximum grades.
4. Create a Main class to test the Student class:
o Instantiate several Student objects.
o Call printMaxGrades() on each object.
o Call displaySchoolName() on each object.
o Print the total number of students using the totalStudents variable

sol:

```
class Student{
    final int MAX_GRADES=10;
    static int totalStudents=0;
    String StudentName;
    String schoolName="Akshara Medium School";

    final void printMaxGrades() {
        System.out.println("Grade :"+MAX_GRADES);
    }

    static void incrementStudentCount() {
        totalStudents++;
    }
    void displaySchoolName() {
        System.out.println("School Name :"+schoolName);
    }

    public Student(String StudentName) {
        this.StudentName=StudentName;
        incrementStudentCount();
        displaySchoolName();
    }

    @Override
    public String toString() {
        return "Student [MAX_GRADES=" + MAX_GRADES + ", StudentName=" + StudentName + " ]";
    }

    public static void main(String [] args) {
        Student std1=new Student("Manikanta");
        System.out.println(std1);
        System.out.println();
        Student std2=new Student("Pavan");
        System.out.println(std2);
        System.out.println();
        Student std3=new Student("Srinu");
        System.out.println(std3);
        System.out.println();
    }
}
```

```

        System.out.println("Total number of Students :"+totalStudents);
    }
}

```

Assignment 3: Bank Account System

Objective

Apply final, static, and global variables and methods in the context of managing bank accounts.

Instructions

1. Create a Java class named BankAccount.

2. Define the following:

- o A final variable INTEREST_RATE representing the interest rate applied to accounts.

Set this to 0.03 (3%).

- o A static variable totalAccounts that keeps track of the number of BankAccount instances created.

- o A global (class-level) variable bankName that specifies the name of the bank.

- o A final method printInterestRate() that prints the interest rate.

- o A static method incrementAccountCount() that increments the totalAccounts variable.

- o A non-static method displayBankName() that prints the bank name.

3. In the BankAccount class:

- o Create a constructor that initializes the account holder's name and balance, and

increments the account count using the incrementAccountCount() method.

- o Override the toString() method to provide information about the bank account, including the holder's name and balance.

4. Create a Main class to test the BankAccount class:

- o Instantiate several BankAccount objects.

- o Call printInterestRate() on each object.

- o Call displayBankName() on each object.

- o Print the total number of bank accounts using the totalAccounts variable.

sol:

```

public class BankAccount {
    final float INTEREST_RATE=0.03f;
    static int totalAccounts=0;
    String bankName="UNION BANK";
    String accountHolderName;
    double balance;

    final void printInterestRate() {
        System.out.println("Interest Rate :"+INTEREST_RATE);
    }

    static void incrementAccountCount() {
        totalAccounts++;
    }
}

```

```

void displayBankName() {
    System.out.println("Bank Name :"+bankName);
}

public BankAccount(String accountHolderName, double balance) {

    this.accountHolderName = accountHolderName;
    this.balance = balance;
    displayBankName();
    incrementAccountCount();
    printInterestRate();
}

public String toString() {
    return "BankAccount [accountHolderName=" + accountHolderName +
", balance=" + balance + "]\n";
}

public static void main(String[] args) {
    BankAccount ba1=new BankAccount("UNION BANK",100000);
    System.out.println(ba1);
    System.out.println();

    BankAccount ba2=new BankAccount("UNION BANK",50000);
    System.out.println(ba2);
    System.out.println();

    BankAccount ba3=new BankAccount("UNION BANK",90000);
    System.out.println(ba3);
    System.out.println();
    System.out.println("Total number of Bank Accounts
:"+totalAccounts);

}
}

```

Assignment 4: Employee Management System

Objective

Explore final, static, and global variables and methods in managing employee data.

Instructions

1. Create a Java class named Employee.
2. Define the following:
 - o A final variable MAX_VACATION_DAYS that represents the maximum number of vacation days an employee can accrue. Set this to 20.
 - o A static variable totalEmployees to keep track of the number of Employee instances created.
 - o A global (class-level) variable companyName that specifies the name of the company.
 - o A final method printMaxVacationDays() that prints the maximum number of

vacation
days.

- o A static method incrementEmployeeCount() that increments the totalEmployees variable.

- o A non-static method displayCompanyName() that prints the company name.

3. In the Employee class:

- o Create a constructor that initializes the employee's name and position, and increments the employee count using the incrementEmployeeCount() method.

- o Override the toString() method to provide information about the employee, including their name and position.

4. Create a Main class to test the Employee class:

- o Instantiate several Employee objects.

- o Call printMaxVacationDays() on each object.

- o Call displayCompanyName() on each object.

- o Print the total number of employees using the totalEmployees variable.

sol:

```
class Employee{
    final int MAX_VACATION_DAYS=20;
    static int totalEmployees=0;
    String companyName="HCL";
    String empName;
    String position;

    final void printMaxVacationDays() {
        System.out.println("Total vacation holidays
:"+MAX_VACATION_DAYS);
    }
    static void incrementEmployeeCount() {
        totalEmployees++;
    }
    void displayCompanyName() {
        System.out.println("Company Name :"+companyName);
    }
    public Employee(String EmpName,String Position) {
        this.empName = EmpName;
        this.position=Position;
        incrementEmployeeCount();
        displayCompanyName();
        printMaxVacationDays();
    }

    @Override
    public String toString() {
        return "Employee [empName=" + empName + ", position=" + position
+ " ]";
    }
    public static void main(String[] args) {
        Employee emp1=new Employee("Manikanta","Manager");
        System.out.println(emp1);
    }
}
```

```

        System.out.println();
        Employee emp2=new Employee("Pavan","Teamlead");
        System.out.println(emp2);
        System.out.println();
        Employee emp3=new Employee("Srinu","HR");
        System.out.println(emp3);
        System.out.println();
        System.out.println("Total number of Employees
:"+totalEmployees);
    }

}

```

Assignment 5: E-commerce Product Management

Objective

Apply final, static, and global variables and methods in a product management context.

Instructions

1. Create a Java class named Product.
2. Define the following:
 - o A final variable TAX_RATE that represents the sales tax rate. Set this to 0.08 (8%).
 - o A static variable totalProducts that keeps track of the number of Product instances.
 - o A global (class-level) variable storeName that specifies the name of the store.
 - o A final method printTaxRate() that prints the tax rate.
 - o A static method incrementProductCount() that increments the totalProducts variable.
 - o A non-static method displayStoreName() that prints the store name.
3. In the Product class:
 - o Create a constructor that initializes the product name and price, and increments the product count using the incrementProductCount() method.
 - o Override the toString() method to provide information about the product, including its name and price.
4. Create a Main class to test the Product class:
 - o Instantiate several Product objects.
 - o Call printTaxRate() on each object.
 - o Call displayStoreName() on each object.
 - o Print the total number of products using the totalProducts variable

sol:

```

public class Product {
    final float TAX_RATE=(float) 0.08;
    static int totalProducts=0;
    String storeName="Prabha Medicals";
    String product;
    int productPrice;
}

```

```

final void printTaxRate() {
    System.out.println("Tax :"+TAX_RATE);
}
static void incrementProductCount() {
    totalProducts++;
}
void displayStoreName() {
    System.out.println("Store Name :"+storeName);
}
public Product(String Product,int ProductPrice) {
    this.product = Product;
    this.productPrice=ProductPrice;
    incrementProductCount();
    printTaxRate();
    displayStoreName();
}

```

```

@Override
public String toString() {
    return "Product [product=" + product + ", productPrice=" +
productPrice + "]";
}
public static void main(String[] args) {
    Product pt1=new Product("Dolo",2);
    System.out.println(pt1);
    System.out.println();
    Product pt2=new Product("Dart",5);
    System.out.println(pt2);
    System.out.println();
    Product pt3=new Product("Cefemix",10);
    System.out.println(pt3);
    System.out.println();
    System.out.println("Total number of Products :"+totalProducts);
}
}

```