

Q1) Write a program to display values of enums using a constructor & getPrice() method (Example display house & their prices)

```
public class Main {  
    public static void main(String[] args) {  
        House large = House.LARGE;  
        House medium = House.MEDIUM;  
        House small = House.SMALL;  
  
        large.getPrice();  
        small.getPrice();  
        medium.getPrice();  
    }  
}
```

Main x

/home/aman/.jdk/openjdk-24.0.1/bin/java -javaagen
Price of LARGE: 10000
Price of SMALL: 6000
Price of MEDIUM: 8000

Process finished with exit code 0

```
public enum House { 2 usages  
    SMALL(price: 6000), MEDIUM(price: 8000), LARGE(price: 10000); no usages  
  
    private int price; 2 usages  
  
    House(int price) { 6 usages  
        this.price = price;  
    }  
  
    public void getPrice(){ 1 usage  
        System.out.println("Price of " + this.name() + ": " + price);  
    }  
}
```

Q2) Create a User class with fields: firstname, lastname, age, phonenumber. Write a program which accepts values of user fields from commandline, create object and append that to a text file. After every user creation the program should prompt: "Do you want to continue creating users? (Type QUIT to exit)" and keep on accepting values and writing to file until user quits.

```
public class User { no usages
    String firstName; 2 usages
    String lastName; 2 usages
    int age; 2 usages
    long phoneNumber; 2 usages

    public User(String firstName, String lastName, int age, long phoneNumber) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
        this.phoneNumber = phoneNumber;
    }

    public String toString() {
        return firstName + " " + lastName + " " + age + " " + " " + phoneNumber;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try(BufferedWriter bc = new BufferedWriter(new
FileWriter("Test.txt"))){
            while(true){
                System.out.println("Enter firstName :");
                String firstName = sc.nextLine();

                System.out.println("Enter lastName :");
                String lastName = sc.nextLine();

                System.out.println("Enter age :");
                int age = Integer.parseInt(sc.nextLine());
```

```

        System.out.println("Enter phoneNumber :");
        long phoneNumber = Long.parseLong(sc.nextLine());
        System.out.println();

        User user = new User(firstName, lastName, age,
phoneNumber);

        bc.write(user.toString() + '\n');
        bc.flush();

        System.out.println("DO YOU WANT TO CONTINUE? TO QUIT
ENTER \"QUIT\"");
        String line = sc.nextLine();

        if(line.contains("QUIT")){
            break;
        }
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
} finally {
    System.out.println("Program closed successfully");
}

}
}

```

Q3)Write a program to count number of occurrences of a word in a file. The file name and word should be supplied through commandline.

```

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter file name: ");
        String fileName = sc.nextLine();

        System.out.print("Enter the word to search: ");
        String searchWord = sc.nextLine();
    }
}

```

```

        int count = 0;

        try(BufferedReader bc = new BufferedReader(new
FileReader(fileName))){
            String line;
            while((line=bc.readLine())!=null){
                String[] Word = line.split("\\W+");

                for(String word: Word){
                    if(word.equals(searchWord)){
                        count++;
                    }
                }
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        } finally {
            System.out.println("Word is found " + count + " times.T");
        }
    }
}

```

Output

```

/home/aman/.jdk/openjdk-24.0.1/bin/java -java
Enter file name: Test.txt
Enter the word to search: see
Word is found 11 times.

Process finished with exit code 0

```

Q4)Write a program to show application of Factory Design Pattern.

```
// Abstract Product Class
abstract class Document {
    public abstract void open();
}

// Concrete Products
class Report extends Document {
    @Override
    public void open() {
        System.out.println("Opening a Report document.");
    }
}

class Spreadsheet extends Document {
    @Override
    public void open() {
        System.out.println("Opening a Spreadsheet document.");
    }
}

// Creator Abstract Class
abstract class Application {
    public abstract Document createDocument();
}

// Concrete Creators
class ReportApplication extends Application {
    @Override
    public Document createDocument() {
        return new Report();
    }
}

class SpreadsheetApplication extends Application {
    @Override
    public Document createDocument() {
        return new Spreadsheet();
    }
}

// Client Code
public class DocumentCreatorExample {
```

```
public static void main(String[] args) {  
    Application reportApp = new ReportApplication();  
    Document report = reportApp.createDocument();  
    report.open();  
  
    Application spreadsheetApp = new SpreadsheetApplication();  
    Document spreadsheet = spreadsheetApp.createDocument();  
    spreadsheet.open();  
}  
}
```

Q5)Write a program to show application of Singleton Design Pattern.

```
public class Student {  
    private static Student student;  
  
    private Student() {  
  
    }  
  
    public static Student getInstance() {  
        if (student == null) {  
            student = new Student();  
        }  
        return student;  
    }  
}
```