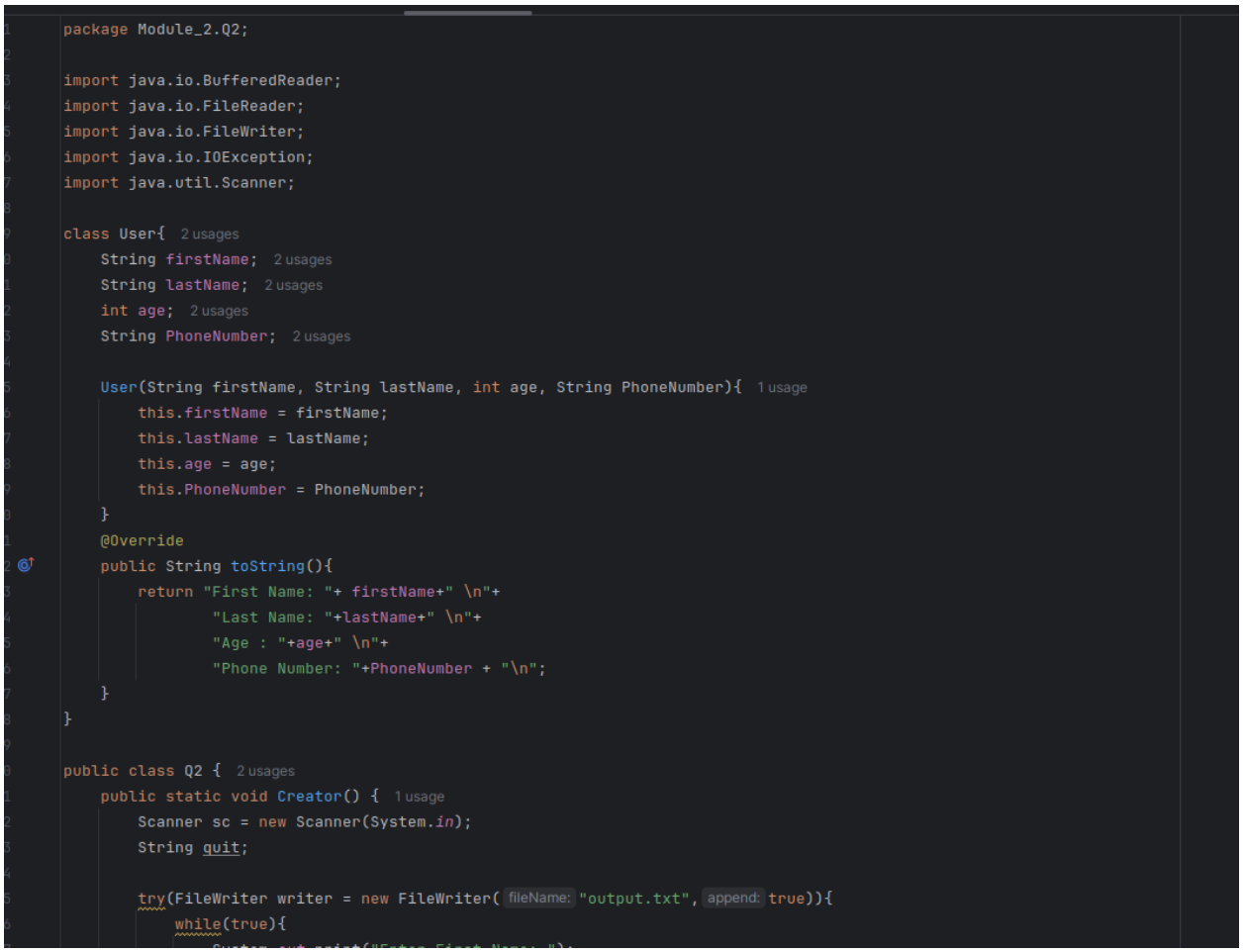
Assignment - Introduction to Java - 2

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

```java
package Module_2.Q1;

public enum House{  no usages
    SMALL( price: 1000),  no usages
    MEDIUM( price: 10000),  no usages
    LARGE( price: 100000);  no usages

    private final int price;  2 usages

    House(int price){  6 usages
        this.price = price;
    }

    public int getPrice(){  no usages
        return price;
    }

}
```

```java
package Module_2.Q1;

public class Q1 {  2 usages
    public static void ENUM() {  1 usage
        House small = House.SMALL;
        House medium = House.MEDIUM;
        House large = House.LARGE;

        System.out.println("House Type :"+ small.name() +"  "+ small.getPrice());
        System.out.println("House Type :"+ medium.name() +"  "+medium.getPrice());
        System.out.println("House Type :"+ large.name() +"  "+large.getPrice());
    }
}
```

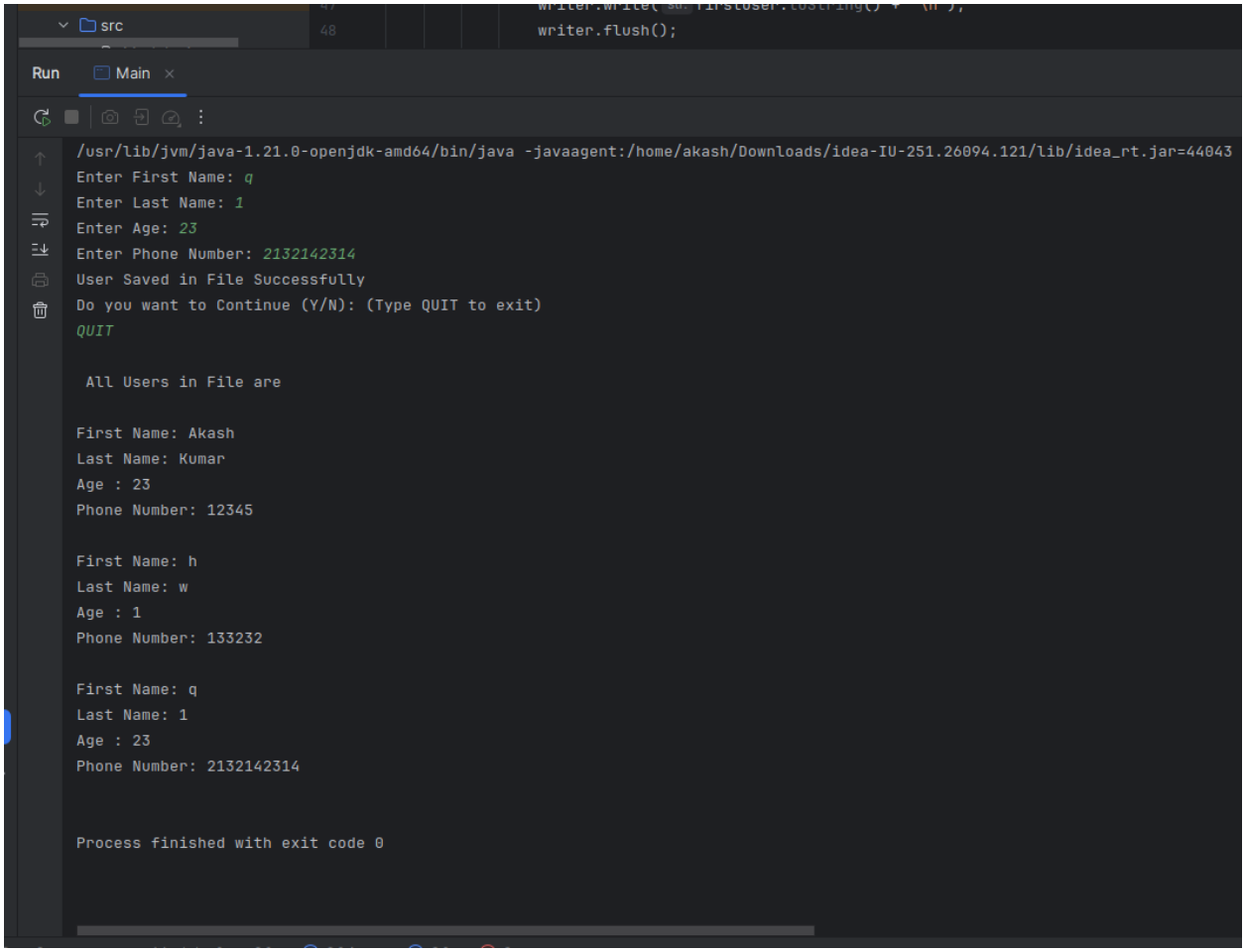
Q2)Create a User class with fields: firstname, lastname, age, phonenumber. Write a program which accepts values of user fields from commandline, create object and append that to a text file. After every user creation the program should prompt: "Do you want to continue creating users? (Type QUIT to exit)"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



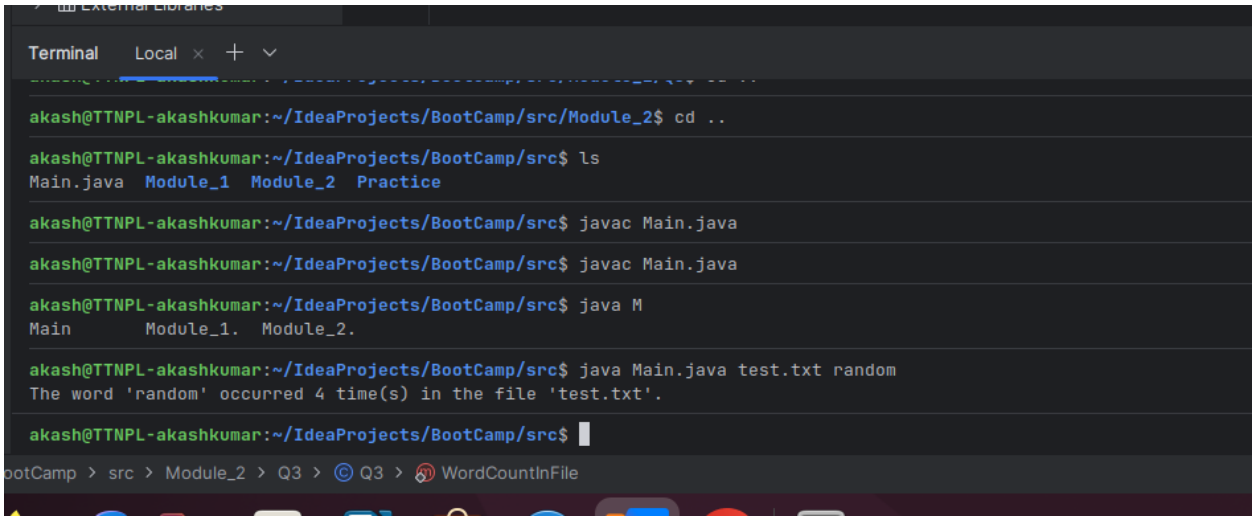
```java
package Module_2.Q2;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

class User{  2 usages
    String firstName;  2 usages
    String lastName;  2 usages
    int age;  2 usages
    String PhoneNumber;  2 usages

    User(String firstName, String lastName, int age, String PhoneNumber){  1 usage
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
        this.PhoneNumber = PhoneNumber;
    }
    @Override
    public String toString(){
        return "First Name: "+ firstName+" \n"+
                "Last Name: "+lastName+" \n"+
                "Age : "+age+" \n"+
                "Phone Number: "+PhoneNumber + "\n";
    }
}

public class Q2 {  2 usages
    public static void Creator() {  1 usage
        Scanner sc = new Scanner(System.in);
        String quit;

        try(FileWriter writer = new FileWriter( fileName: "output.txt", append: true)){
            while(true){
```

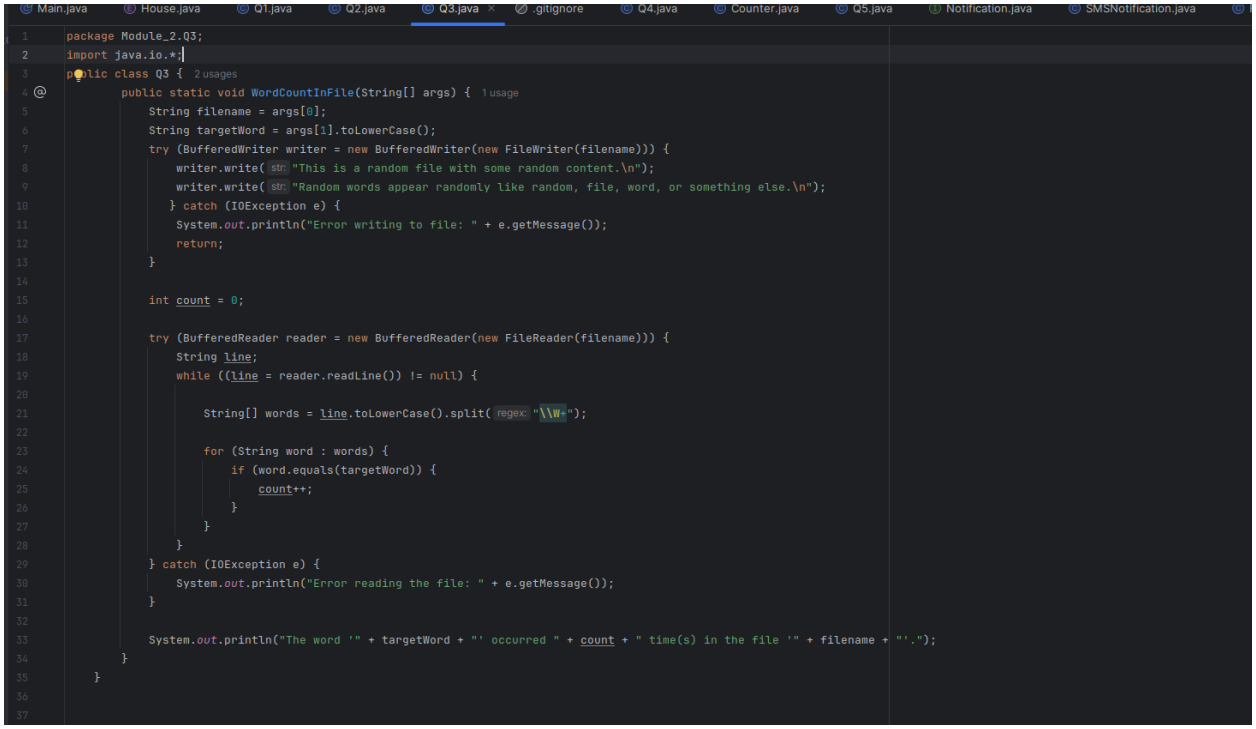
```java
public class Q2 {  2 usages
    public static void Creator() {  1 usage
        String quit;

        try(FileWriter writer = new FileWriter( fileName: "output.txt", append: true)){
            while(true){
                System.out.print("Enter First Name: ");
                String firstName = sc.nextLine();
                System.out.print("Enter Last Name: ");
                String lastName = sc.nextLine();
                System.out.print("Enter Age: ");
                int age = Integer.parseInt(sc.nextLine());
                System.out.print("Enter Phone Number: ");
                String PhoneNumber = sc.nextLine();
                User firstUser = new User(firstName, lastName, age, PhoneNumber);

                writer.write( str: firstUser.toString() + "\n");
                writer.flush();

                System.out.print("User Saved in File Successfully \n");

                System.out.println("Do you want to Continue (Y/N): (Type QUIT to exit)");
                quit = sc.nextLine();
                if(quit.equals("QUIT")){
                   break;
                }
            }

            System.out.print("\n All Users in File are \n");
            try(BufferedReader br = new BufferedReader(new FileReader( fileName: "output.txt"))){
                String line;
                while((line = br.readLine()) != null){
                    System.out.println(line);

                }
            }
        } catch (IOException e) {
```

```
public class Q2 {  2 usages
    public static void Creator() {  1 usage

                writer.write( str: firstUser.toString() + "\n");
                writer.flush();

                System.out.print("User Saved in File Successfully \n");

                System.out.println("Do you want to Continue (Y/N): (Type QUIT to exit)");
                quit = sc.nextLine();
                if(quit.equals("QUIT")){
                   break;
                }
            }

            System.out.print("\n All Users in File are \n");
            try(BufferedReader br = new BufferedReader(new FileReader( fileName: "output.txt"))){
                String line;
                while((line = br.readLine()) != null){
                    System.out.println(line);

                }
            }
        } catch (IOException e) {
            throw new RuntimeException(e);
        } finally {

            sc.close();
        }
    }
}
```


src
48
writer.flush();
Run
Main
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/akash/Downloads/idea-IU-251.26094.121/lib/idea_rt.jar=44043
Enter First Name: q
Enter Last Name: 1
Enter Age: 23
Enter Phone Number: 2132142314
User Saved in File Successfully
Do you want to Continue (Y/N): (Type QUIT to exit)
QUIT
All Users in File are
First Name: Akash
Last Name: Kumar
Age : 23
Phone Number: 12345
First Name: h
Last Name: w
Age : 1
Phone Number: 133232
First Name: q
Last Name: 1
Age : 23
Phone Number: 2132142314
Process finished with exit code 0

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.





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

```java
package Module_2.Q4;

public interface Notification {  5 usages  2 implementations
    void notifyuser();  no usages  2 implementations
}
```

```java
package Module_2.Q4;

public class EmailNotification implements Notification {  1 usage
    @Override  1 usage
    public void notifyuser() {
        System.out.println("Email Notification");
    }
}
```

```java
package Module_2.Q4;

public class SMSNotification implements Notification {  1 usage
    @Override  2 usages
    public void notifyuser() {
        System.out.println("SMS Notification");
    }
}
```

```java
package Module_2.Q4;

public class FactoryNotification {  2 usages
    Change signature
    public Notification createNotification(String type) {  2 usages
        if(type == null || type.isEmpty())
        {
            return null;
        }
        if(type.equalsIgnoreCase( anotherString: "SMS")){
            return new SMSNotification();
        } else if(type.equalsIgnoreCase( anotherString: "Email")){
            return new EmailNotification();
        }
        return null;
    }
}
```

```java
package Module_2.Q4;


public class Q4 {  1 usage
    public static void Factory(){  1 usage
        FactoryNotification factory = new FactoryNotification();
        Notification not1 = factory.createNotification( type: "Email");
        not1.notifyuser();
        Notification not2 = factory.createNotification( type: "SMS");
        not2.notifyuser();
    }
}

```

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/akash/Downloads/idea-IU-251.26094.1
Email Notification
SMS Notification

Process finished with exit code 0
```

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

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/home/akash/Downloads/idea-IU-251.26094.1
Count from c1: 3
Count from c2: 3
c1 and c2 are the same instance (Singleton).

Process finished with exit code 0
```

```java
public class Counter {  7 usages
    private static Counter instance;  3 usages

    private int count = 0;  2 usages

    private Counter() {}  1 usage

    public static Counter getInstance() {  2 usages
        if (instance == null) {
            instance = new Counter();
        }
        return instance;
    }

    public void increment() {  3 usages
        count++;
    }

    public int getCount() {  2 usages
        return count;
    }
}
```

```java
package Module_2.Q5;

public class Q5 {  2 usages
    public static void Singleton() {  1 usage
        Counter c1 = Counter.getInstance();
        Counter c2 = Counter.getInstance();

        c1.increment();
        c1.increment();

        c2.increment();

        System.out.println("Count from c1: " + c1.getCount());
        System.out.println("Count from c2: " + c2.getCount());

        if (c1 == c2) {
            System.out.println("c1 and c2 are the same instance (Singleton).");
        }
    }

}
```