

Code:

```
import java.io.*;
import java.util.*;

public class JavaLabFat {
    public static void main(String[] args) throws IOException,
    ClassNotFoundException {
        try {
            Scanner sc = new Scanner(System.in);
            Scanner sc1 = new Scanner(System.in);
            System.out.print("Enter number of mobile phones: ");
            int n = sc.nextInt();
            mobile marr[] = new mobile[n];
            File obj = new File("/home/subham/Desktop/JAVA LAB
FAT/mobilephone.txt");
            FileOutputStream fout = new FileOutputStream(obj);
            ObjectOutputStream objout = new ObjectOutputStream(fout);
            System.out.println();
            for (int i = 0; i < n; i++) {
                System.out.println("-----ENTER DETAILS OF Mobile " + (i +
1) + " -----");
                System.out.print("Enter name of model: ");
                String modelName = sc1.nextLine();
                System.out.print("Enter number of model: ");
                String modelNumber = sc1.nextLine();
                System.out.print("Enter name of brand: ");
                String brandName = sc1.nextLine();
                System.out.print("Enter price of model: ");
                int price = sc.nextInt();
                System.out.print("Enter quantity of the model available:
");
                int quantityAvailable = sc.nextInt();
                marr[i] = new mobile(modelName, modelNumber, brandName,
price, quantityAvailable);
                objout.writeObject(marr[i]);
                System.out.println();
            }
            objout.close();
            fout.close();
        }
    }
}
```

```
        System.out.println();

        FileInputStream fin = new FileInputStream(obj);
        ObjectInputStream objin = new ObjectInputStream(fin);
        mobile mInputArr[] = new mobile[n];
        for (int i = 0; i < n; i++) {
            mInputArr[i] = (mobile) objin.readObject();
        }
        objin.close();
        fin.close();

        System.out.println("-----SELECT AN OPTION-----");
        System.out.println("1.View details of mobile phone by brand
name");
        System.out.println("2.View details of mobile phones within a
price range");
        System.out.println("3.View the brand that has highest number of
models with price less than Rs.10000/-");
        System.out.print("Enter you choice: ");
        int ch = sc.nextInt();
        switch (ch) {
            case 1:
                mobile.viewDetailsByBrandName(mInputArr);
                break;
            case 2:
                mobile.viewDetailsWithinPriceRange(mInputArr);
                break;
            case 3:
                mobile.viewBrandsWithHighestModelsLessThan10k(mInputArr);
                break;
            default:
                System.out.println("Enter a valid option");
        }
        sc.close();
        scl.close();
    } catch (InputMismatchException e) {
        System.out.println("Please eneter the input of correct type");
    }
}
```

```
        } catch (Exception e) {
            e.printStackTrace();
        }

    }
}

class mobile implements Serializable {
    String modelName;
    String modelNumber;
    String brandName;
    int price;
    int quantityAvailable;

    public mobile(String modelName, String modelNumber, String brandName,
int price, int quantityAvailable) {
        this.modelName = modelName;
        this.modelNumber = modelNumber;
        this.brandName = brandName;
        this.price = price;
        this.quantityAvailable = quantityAvailable;
    }

    public void viewDetails() {
        System.out.println("Model Name: " + modelName + ", Model Number: "
+ modelNumber + ", Brand Name: " + brandName
        + ", Price: " + price + ", Quantity Available: " +
quantityAvailable);
    }

    public static void viewDetailsByBrandName(mobile[] marr) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter name of brand: ");
        String brand = sc.nextLine();
        int flag = 0;
        for (mobile m : marr) {
            if (m.brandName.compareToIgnoreCase(brand) == 0) {
                m.viewDetails();
                flag = 1;
            }
        }
    }
}
```

```
    }

    }

    if (flag == 0) {
        System.out.println("No mobiles of the brand were found");
    }
    sc.close();
}

public static void viewDetailsWithinPriceRange(mobile[] marr) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter lower range of the price: ");
    int low = sc.nextInt();
    System.out.print("Enter higher range of the price: ");
    int high = sc.nextInt();
    int flag = 0;
    for (mobile m : marr) {
        if (m.price >= low && m.price <= high) {
            m.viewDetails();
            flag = 1;
        }
    }
    if (flag == 0) {
        System.out.println("No mobiles in that price range were
found");
    }
    sc.close();
}

public static void viewBrandsWithHighestModelsLessThan10k(mobile
marr[]) {
    HashMap<String, Integer> mobileCount = new HashMap<>();
    int flag = 0;
    for (mobile m : marr) {
        if (m.price <= 10000) {
            if (mobileCount.containsKey(m.brandName)) {
                mobileCount.replace(m.brandName,
mobileCount.get(m.brandName).intValue() + 1);
            } else {
```

```
        mobileCount.put(m.brandName, 1);
        flag = 1;
    }
}

if (flag == 0) {

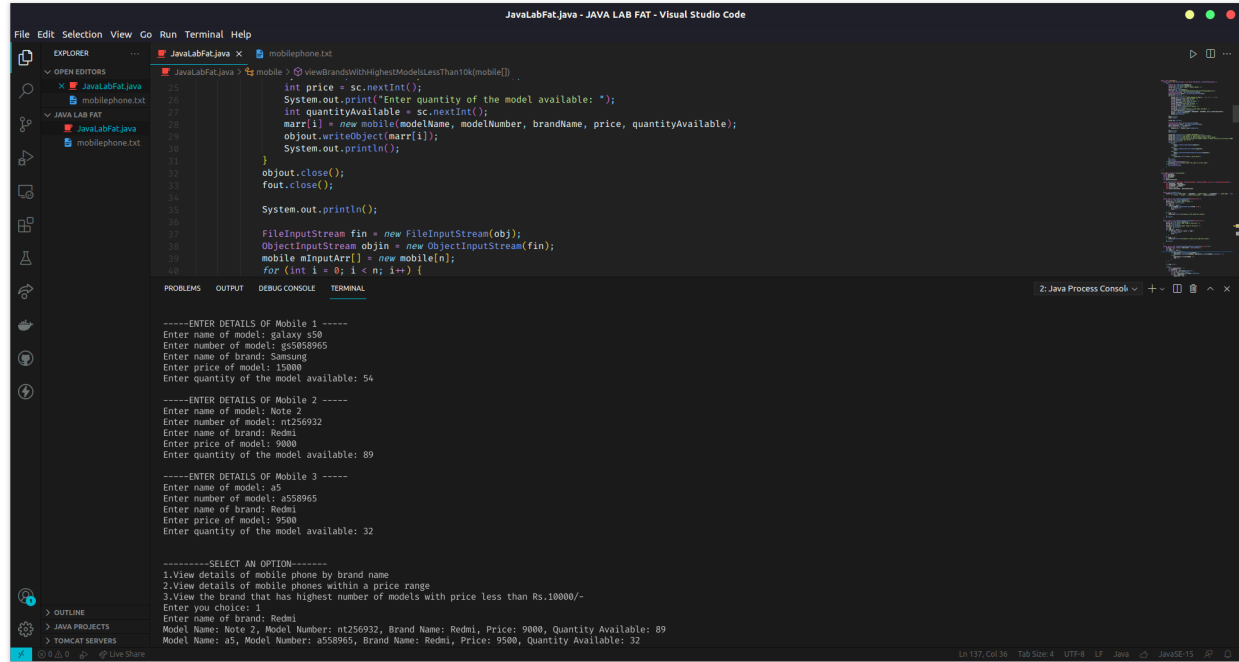
} else {
    String highestBrand = "";
    int highestVal = 0;
    for (String b : mobileCount.keySet()) {
        String brand = b.toString();
        int val = mobileCount.get(brand).intValue();
        if (val > highestVal) {
            highestVal = val;
            highestBrand = brand;
        }
    }
    System.out.println(
        "The brand that has the highest number of Models with
price lesser than Rs.10000: " + highestBrand);
}

}
```

19BIT0093 - Subham Subhasish Panda

JAVA LAB FAT

Editor Screenshot with Code and Output (IDE used - VSCODE):



The screenshot displays the Visual Studio Code IDE with the file `JavaLabFat.java` open. The code defines a `mobile` class with attributes `modelName`, `modelNumber`, `brandName`, `price`, and `quantityAvailable`. It includes methods for inputting details, displaying details, and selecting an option. The `main` method uses a `FileInputStream` to read input from `mobile.txt` and processes it based on the selected option.

```
JavaLabFat.java - JAVA LAB FAT - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
  OPEN EDITORS
    JavaLabFat.java x
    mobilephone.txt
  JAVA LAB FAT
    JavaLabFat.java
    mobilephone.txt

JavaLabFat.java
25 int price = sc.nextInt();
26 System.out.print("Enter quantity of the model available: ");
27 int quantityAvailable = sc.nextInt();
28 marz[i] = new mobile(modelName, modelNumber, brandName, price, quantityAvailable);
29 objout.writeObject(marr[i]);
30 System.out.println();
31 }
32 objout.close();
33 fout.close();
34 }
35 System.out.println();
36 }
37 FileInputStream fin = new FileInputStream(obj);
38 ObjectInputStream objin = new ObjectInputStream(fin);
39 mobile mInputArr[] = new mobile[n];
40 for (int i = 0; i < n; i++) {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Java Process Console

-----ENTER DETAILS OF Mobile 1 -----
Enter name of model: galaxy s50
Enter number of model: gs508965
Enter name of brand: Samsung
Enter price of model: 15000
Enter quantity of the model available: 54

-----ENTER DETAILS OF Mobile 2 -----
Enter name of model: Note 2
Enter number of model: nt256932
Enter name of brand: Redmi
Enter price of model: 9800
Enter quantity of the model available: 89

-----ENTER DETAILS OF Mobile 3 -----
Enter name of model: a5
Enter number of model: a558965
Enter name of brand: Redmi
Enter price of model: 9500
Enter quantity of the model available: 32

-----SELECT AN OPTION-----
1.View details of mobile phone by brand name
2.View details of mobile phones within a price range
3.View the brand that has highest number of models with price less than Rs.10000/-
Enter you choice: 1
Enter name of brand: Redmi
Model Name: Note 2, Model Number: nt256932, Brand Name: Redmi, Price: 9800, Quantity Available: 89
Model Name: a5, Model Number: a558965, Brand Name: Redmi, Price: 9500, Quantity Available: 32
```

19BIT0093 - Subham Subhasish Panda

JAVA LAB FAT

Output:

1. Details of mobile phone by brand name:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
| | ~/Desktop/JAVA LAB FAT  cd "/home/subham/Desktop/JAVA LAB FAT" ; /usr/bin/env /usr/lib/jvm/jdk-15.
e/User/workspaceStorage/925f40ab8aacbdbb9ac05132b553d420/redhat.java/jdt_ws/JAVA LAB FAT_e9dfa2d5/bin"
Enter number of mobile phones: 3

-----ENTER DETAILS OF Mobile 1 -----
Enter name of model: galaxy s50
Enter number of model: gs5058965
Enter name of brand: Samsung
Enter price of model: 15000
Enter quantity of the model available: 54

-----ENTER DETAILS OF Mobile 2 -----
Enter name of model: Note 2
Enter number of model: nt256932
Enter name of brand: Redmi
Enter price of model: 9000
Enter quantity of the model available: 89

-----ENTER DETAILS OF Mobile 3 -----
Enter name of model: a5
Enter number of model: a558965
Enter name of brand: Redmi
Enter price of model: 9500
Enter quantity of the model available: 32

-----SELECT AN OPTION-----
1.View details of mobile phone by brand name
2.View details of mobile phones within a price range
3.View the brand that has highest number of models with price less than Rs.10000/-
Enter you choice: 1
Enter name of brand: Redmi
Model Name: Note 2, Model Number: nt256932, Brand Name: Redmi, Price: 9000, Quantity Available: 89
Model Name: a5, Model Number: a558965, Brand Name: Redmi, Price: 9500, Quantity Available: 32
```

19BIT0093 - Subham Subhasish Panda

JAVA LAB FAT

2. Details of mobile phone within price range

```
~/Desktop/JAVA LAB FAT cd "/home/subham/Desktop/JAVA LAB FAT" ; /usr/bin/env /usr/lib/jvm/jdk-15.0.2/bin/java -XX:+ShowCodeDetails
e/User/workspaceStorage/925f40ab8aacdbb9ac05132b553d420/redhat.java/jdt_ws/JAVA LAB FAT_e9dfa2d5/bin" JavaLabFat
Enter number of mobile phones: 4

-----ENTER DETAILS OF Mobile 1 -----
Enter name of model: galaxyS50
Enter number of model: gs5058965
Enter name of brand: Samsung
Enter price of model: 15000
Enter quantity of the model available: 58

-----ENTER DETAILS OF Mobile 2 -----
Enter name of model: note 2
Enter number of model: nt259685
Enter name of brand: Redmi
Enter price of model: 9500
Enter quantity of the model available: 57

-----ENTER DETAILS OF Mobile 3 -----
Enter name of model: a3
Enter number of model: a385965
Enter name of brand: Redmi
Enter price of model: 9000
Enter quantity of the model available: 57

-----ENTER DETAILS OF Mobile 4 -----
Enter name of model: F02s
Enter number of model: f02s58965
Enter name of brand: Samsung
Enter price of model: 8500
Enter quantity of the model available: 87

-----SELECT AN OPTION-----
1.View details of mobile phone by brand name
2.View details of mobile phones within a price range
3.View the brand that has highest number of models with price less than Rs.10000/-
Enter you choice: 2
Enter lower range of the price: 7000
Enter higher range of the price: 12000
Model Name: note 2, Model Number: nt259685, Brand Name: Redmi, Price: 9500, Quantity Available: 57
Model Name: a3, Model Number: a385965, Brand Name: Redmi, Price: 9000, Quantity Available: 57
Model Name: F02s, Model Number: f02s58965, Brand Name: Samsung, Price: 8500, Quantity Available: 87
```


3. Details of brand with highest models less than Rs.10000

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

~/Desktop/JAVA LAB FAT  cd "/home/subham/Desktop/JAVA LAB FAT" ; /usr/bin/env /usr/lib/jvm/jdk-15.0.2/
e/User/workspaceStorage/925f40ab8aacbdbb9ac05132b553d420/redhat.java/jdt_ws/JAVA LAB FAT_e9dfa2d5/bin" Java
Enter number of mobile phones: 4

-----ENTER DETAILS OF Mobile 1 -----
Enter name of model: galaxy s50
Enter number of model: gs5058965
Enter name of brand: Samsung
Enter price of model: 15000
Enter quantity of the model available: 59

-----ENTER DETAILS OF Mobile 2 -----
Enter name of model: Note 2
Enter number of model: nt58965
Enter name of brand: Redmi
Enter price of model: 9500
Enter quantity of the model available: 69

-----ENTER DETAILS OF Mobile 3 -----
Enter name of model: a3
Enter number of model: a385965
Enter name of brand: Redmi
Enter price of model: 9000
Enter quantity of the model available: 89

-----ENTER DETAILS OF Mobile 4 -----
Enter name of model: F02s
Enter number of model: f02s58965
Enter name of brand: Samsung
Enter price of model: 8500
Enter quantity of the model available: 58

-----SELECT AN OPTION-----
1.View details of mobile phone by brand name
2.View details of mobile phones within a price range
3.View the brand that has highest number of models with price less than Rs.10000/-
Enter you choice: 3
The brand that has the highest number of Models with price lesser than Rs.10000: Redmi
```