school management system

```
1   import java.io.*;
2   import java.util.*;
3   import java.text.*;
4   import java.math.*;
5   import java.util.regex.*;
6   import java.util.ArrayList;
7   import java.util.Comparator;
8   import java.util.List;
9   import java.util.stream.Collectors;
10  class Student{
11      private String name;
12      private float percentage;
13      public Student(String name,float percentage)
14      {
15          super();
16          this.name=name;
17          this.percentage=percentage;
18      }
19      public String getName()
20      {
21          return name;
22      }
23      public void setName(String name)
24      {
25          this.name=name;
26
27      }
28      public float getPercentage()
29      {
30          return percentage;
```

Autocomplete reconnecting...

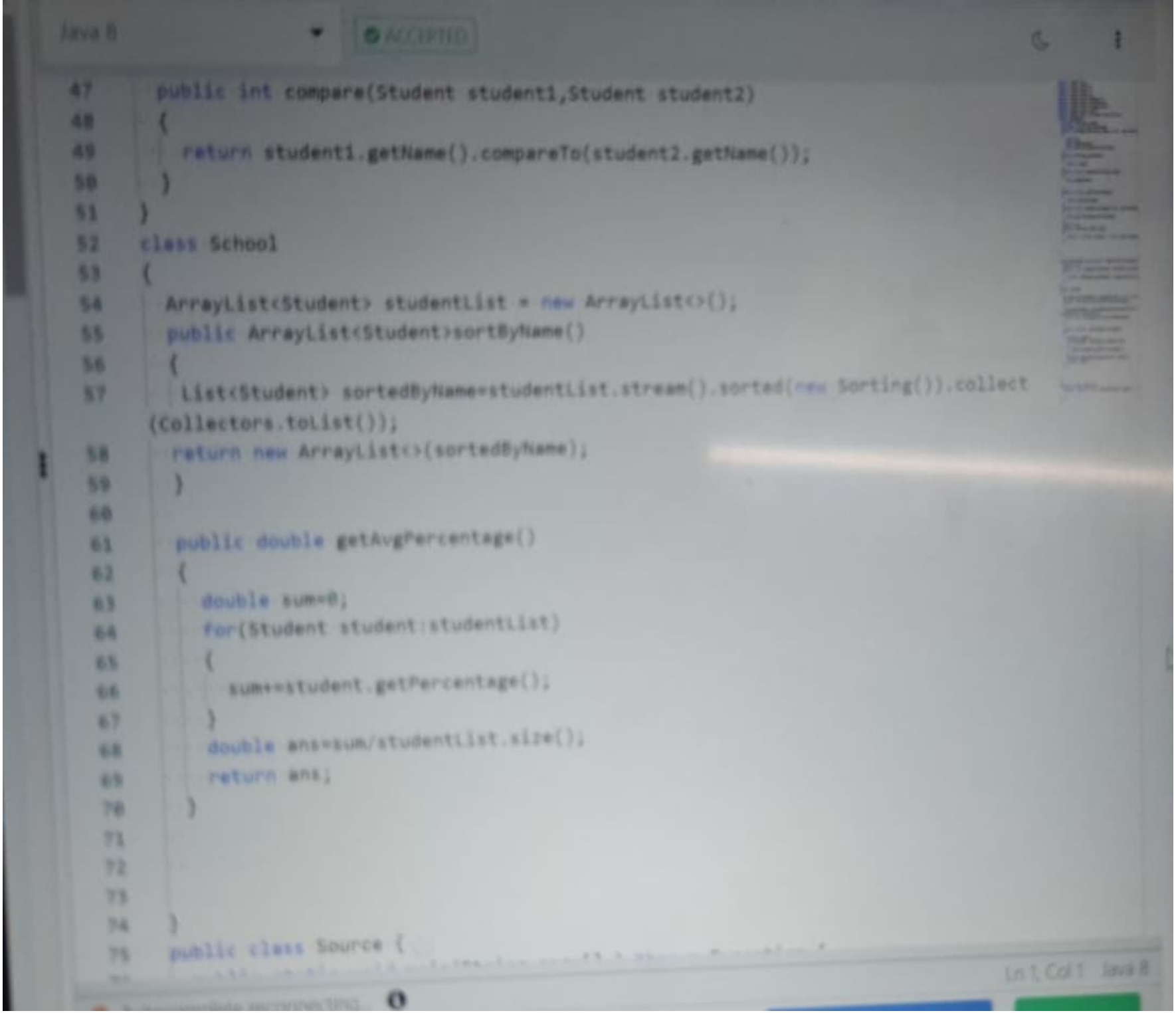
Test Results | Custom Input | RUN COD

Java 8 ACCEPTED

```
28    public float getPercentage()
29    {
30      return percentage;
31    }
32    public void setPercentage(float percentage)
33    {
34      this.percentage=percentage;
35    }
36    @Override
37    public String toString()
38    {
39      return "{"+this.name+","+this.percentage+"}";
40    }
41
42
43  }
44
45  class Sorting implements Comparator<Student>{
46    @Override
47    public int compare(Student student1,Student student2)
48    {
49      return student1.getName().compareTo(student2.getName());
50    }
51  }
52  class School
53  {
54    ArrayList<Student> studentList = new ArrayList<>();
55    public ArrayList<Student>sortByName()
56    {
57      List<Student> sortedByName=studentList.stream().sorted(new Sorting()).collect
```

Autocomplete reconnecting... Ln 1, Col 1 Java 8

Test Results Custom Input RUN CODE SUBMIT



```
47    public int compare(Student student1,Student student2)
48    {
49      return student1.getName().compareTo(student2.getName());
50    }
51  }
52  class School
53  {
54    ArrayList<Student> studentList = new ArrayList<>();
55    public ArrayList<Student>sortByName()
56    {
57      List<Student> sortedByName=studentList.stream().sorted(new Sorting()).collect
    (Collectors.toList());
58    return new ArrayList<>(sortedByName);
59    }
60
61    public double getAvgPercentage()
62    {
63      double sum=0;
64      for(Student student:studentList)
65      {
66        sum+=student.getPercentage();
67      }
68      double ans=sum/studentList.size();
69      return ans;
70    }
71
72
73
74  }
75  public class Source {
```

Java 8

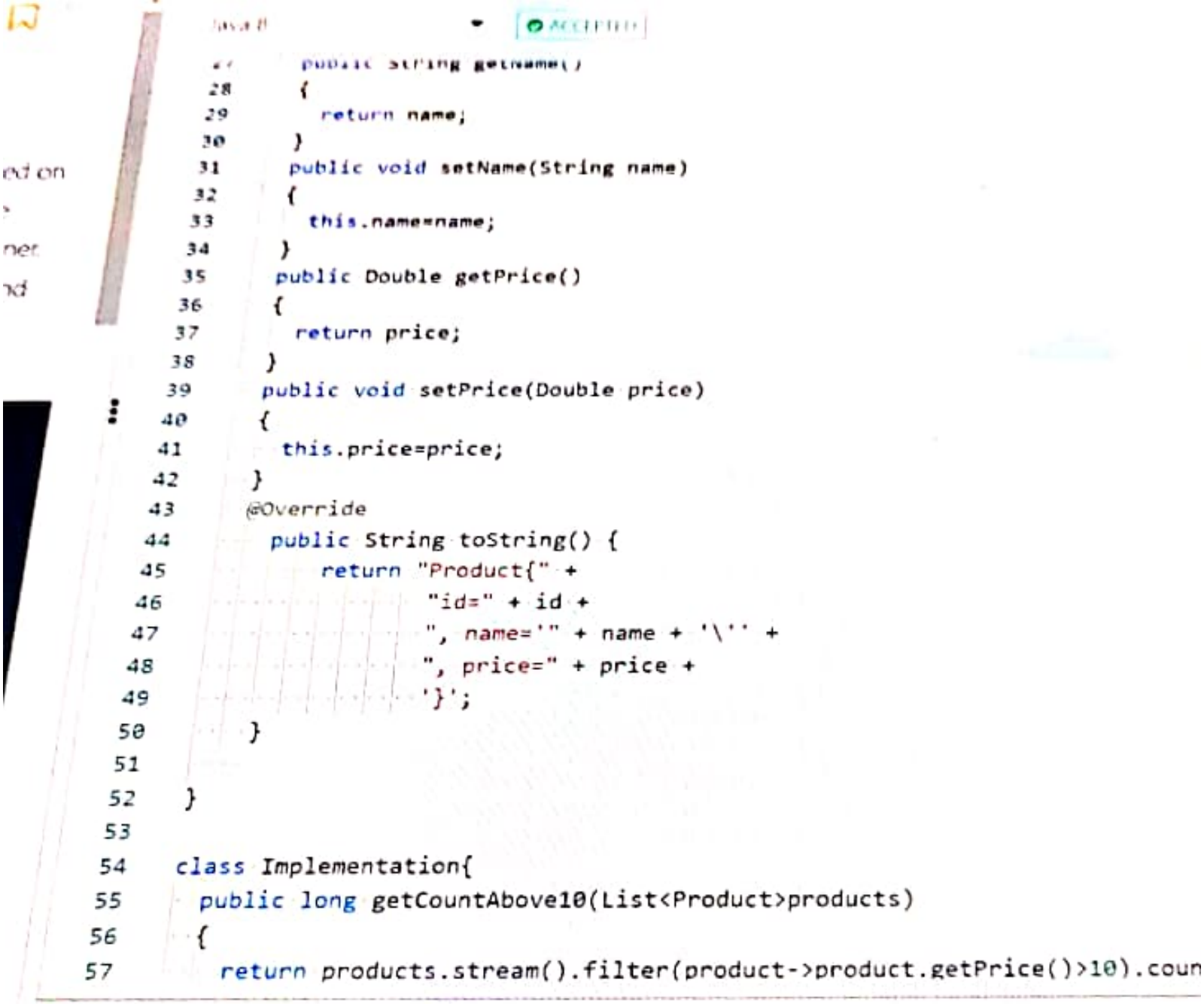
```
1   import java.io.*;
2   import java.util.*;
3   import java.text.*;
4   import java.math.*;
5   import java.util.regex.*;
6   import java.util.ArrayList;
7   import java.util.List;
8
9   class Product {
10      private Integer id;
11      private String name;
12      private Double price;
13      public Product(Integer id,String name,Double price)
14      {
15          this.id=id;
16          this.name=name;
17          this.price=price;
18      }
19      public Integer getId()
20      {
21          return id;
22      }
23      public void setId(Integer id)
24      {
25          this.id=id;
26      }
27      public String getName()
28      {
29          return name;
30      }
```

statistics

Autocomplete reconnecting...

Test Results | Custom Input

RUN CODE



```
28      {
29        return name;
30      }
31      public void setName(String name)
32      {
33        this.name=name;
34      }
35      public Double getPrice()
36      {
37        return price;
38      }
39      public void setPrice(Double price)
40      {
41        this.price=price;
42      }
43      @Override
44        public String toString() {
45            return "Product{" +
46                    "id=" + id +
47                    ", name='" + name + '\'' +
48                    ", price=" + price +
49                    '}';
50      }
51
52  }
53
54  class Implementation{
55    public long getCountAbove10(List<Product>products)
56    {
57      return products.stream().filter(product->product.getPrice()>10).coun
```

```
42      this.price=price;
43  }
44  @Override
45      public String toString() {
46          return "Product{" +
47                  "id=" + id +
48                  ", name='" + name + '\'' +
49                  ", price=" + price +
50                  '}';
51      }
52
53  }
54
55  class Implementation{
56      public long getCountAbove10(List<Product>products)
57      {
58          return products.stream().filter(product->product.getPrice()>10).count();
59      }
60      public double getAverage(List<Product>products)
61      {
62          return products.stream().mapToDouble(product->product.getPrice()).average().
    getAsDouble();
63      }
64  }
65
66  public class Source {
67      public static void main(String args[] ) throws Exception {
68          /* Enter your code here. Read input from STDIN. Print output to STDOUT */
69      }
    }
```

Ln 1, Col 1 Java 8

Autocomplete reconnecting...

RUN CODE

SUBMIT

Java 8

splash uppercase

```
1  import java.io.*;
2  import java.util.*;
3  import java.text.*;
4  import java.math.*;
5  import java.util.regex.*;
6
7  class Implementation
8  {
9    public String spotCapitals(String word)
10   {
11       if(word.toUpperCase().equals(word))
12       {
13           return "CORRECT";
14       }
15       if(word.toLowerCase().equals(word))
16       {
17           return "CORRECT";
18     }
19       String capitalizedWord =word.substring(0,1).toUpperCase()+word.substring(1).
   toLowerCase();
20       if(capitalizedWord.equals(word))
21       {
22           return "CORRECT";
23
24       }
25       return "INCORRECT";
26   }
27 }
28
29 public class Source {
```

Autocomplete reconnecting...

Ln 1, Col 1 Java 8