AOU
الجامعة العربية المفتوحة
Arab Open University
Faculty of Computer Studies

M251: Object Oriented Programming with Java

Tutor-Marked Assignment (TMA) Spring 22/23

Computerized TMA

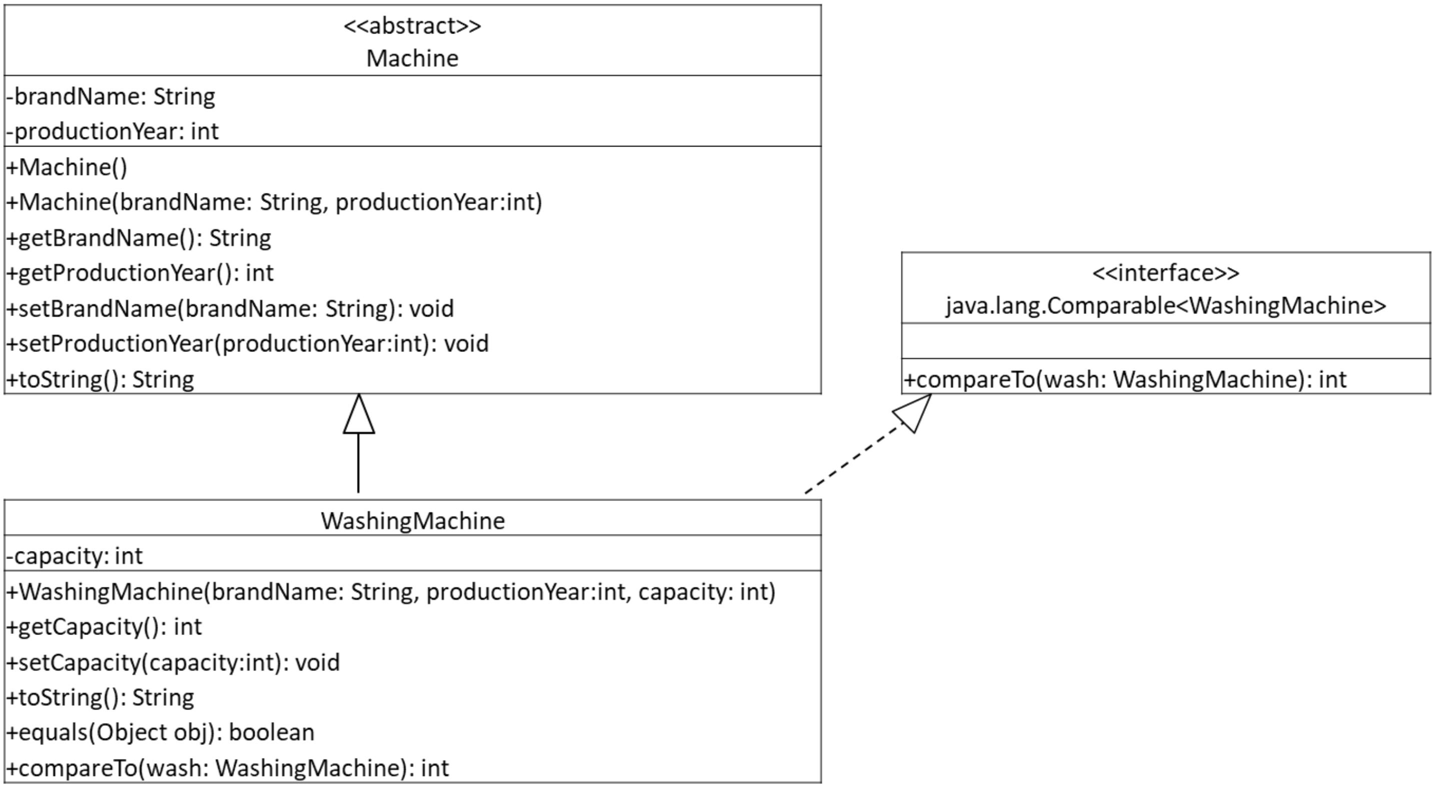
Venue: AOU labs

Total marks: 30 marks turned to 15 marks

Duration: 110 minutes

Sample 1 from the Questions Bank

Consider the following class diagram. Answer the given questions using **Object Oriented Programming** style.



<<abstract>>
Machine

-brandName: String
-productionYear: int

+Machine()
+Machine(brandName: String, productionYear:int)
+getBrandName(): String
+getProductionYear(): int
+setBrandName(brandName: String): void
+setProductionYear(productionYear:int): void
+toString(): String

<<interface>>
java.lang.Comparable<WashingMachine>

+compareTo(wash: WashingMachine): int

WashingMachine

-capacity: int

+WashingMachine(brandName: String, productionYear:int, capacity: int)
+getCapacity(): int
+setCapacity(capacity:int): void
+toString(): String
+equals(Object obj): boolean
+compareTo(wash: WashingMachine): int

Question 1: [7 marks]

Write the code of class **Machine**.

Question 2: [11 marks]

Write the code of class **WashingMachine**.
In **equals()** method, all attributes (including the inherited ones) should be compared.
The comparison in **compareTo()** method is based on **brandName** .

Question 3: [11 marks]

a) Create a text file includes the following data of 6 **WashingMachine** objects, where the values represent brandName, productionYear and capacity, respectively:
Toshiba 2022 6
Samsung 2021 10
Bosch 2023 9
Beko 2022 8
Wansa 2020 9
Beko 2022 8

b) Write a Test class to test classes declared in Q1 and Q2 as follows:

1. Declare and create an ArrayList of **WashingMachine** objects
2. Read data from the text file and store it in the ArrayList. It is not allowed to add the same **WashingMachine** twice and in this case an error message should be displayed.
3. Compute and print on the screen the information of the machine that has the maximum **capacity**.
4. Sort the **WashingMachine** objects in ArrayList according to **brandName.**
5. Print the objects of the sorted ArrayList (one object per line).

Question 4: [1 mark]

Copy and paste the exact output of your code.

End of Questions

Answers:

Q1:

```
//7 marks as in comments
//1
public abstract class Machine {
  //1
  private String brandName;
  private int productionYear;
  //1
  public Machine() { }
  //1
  public Machine(String brandName, int productionYear) {
    this.brandName = brandName;
    this.productionYear = productionYear;
  }
  //2
  public String getBrandName() { return brandName; }
  public void setBrandName(String brandName) { this.brandName = brandName;  }
  public int getProductionYear() { return productionYear; }
  public void setProductionYear(int productionYear) {
    this.productionYear = productionYear;
  }
  //1
  @Override
  public String toString() {
    return "brandName= " + brandName + ", productionYear= " + productionYear;
  }
}
```

Q2:

```
//11 marks as in comments
//2
public class WashingMachine extends Machine implements
Comparable<WashingMachine> {
  //1
  private int capacity;
  //1
  public WashingMachine(String brandName, int productionYear, int capacity) {
    super(brandName, productionYear);
    this.capacity = capacity;
  }
  //1
  public int getCapacity() { return capacity; }
  public void setCapacity(int capacity) { this.capacity = capacity; }
  //2
  @Override
  public String toString() {
    return super.toString() + ", capacity= " + capacity;
  }
  //2
  @Override
  public boolean equals(Object obj) {
    if (this == obj) return true; //optional (not required) checking
    if (!(obj instanceof WashingMachine)) return false;
    WashingMachine that = (WashingMachine) obj;
    return getBrandName().equals(that.getBrandName()) &&
            getProductionYear() == that.getProductionYear() &&
            capacity == that.capacity;
  }
  //2
  @Override
  public int compareTo(WashingMachine wash) {
    return getBrandName().compareTo(wash.getBrandName());
  }
}
```

Q3 (b):

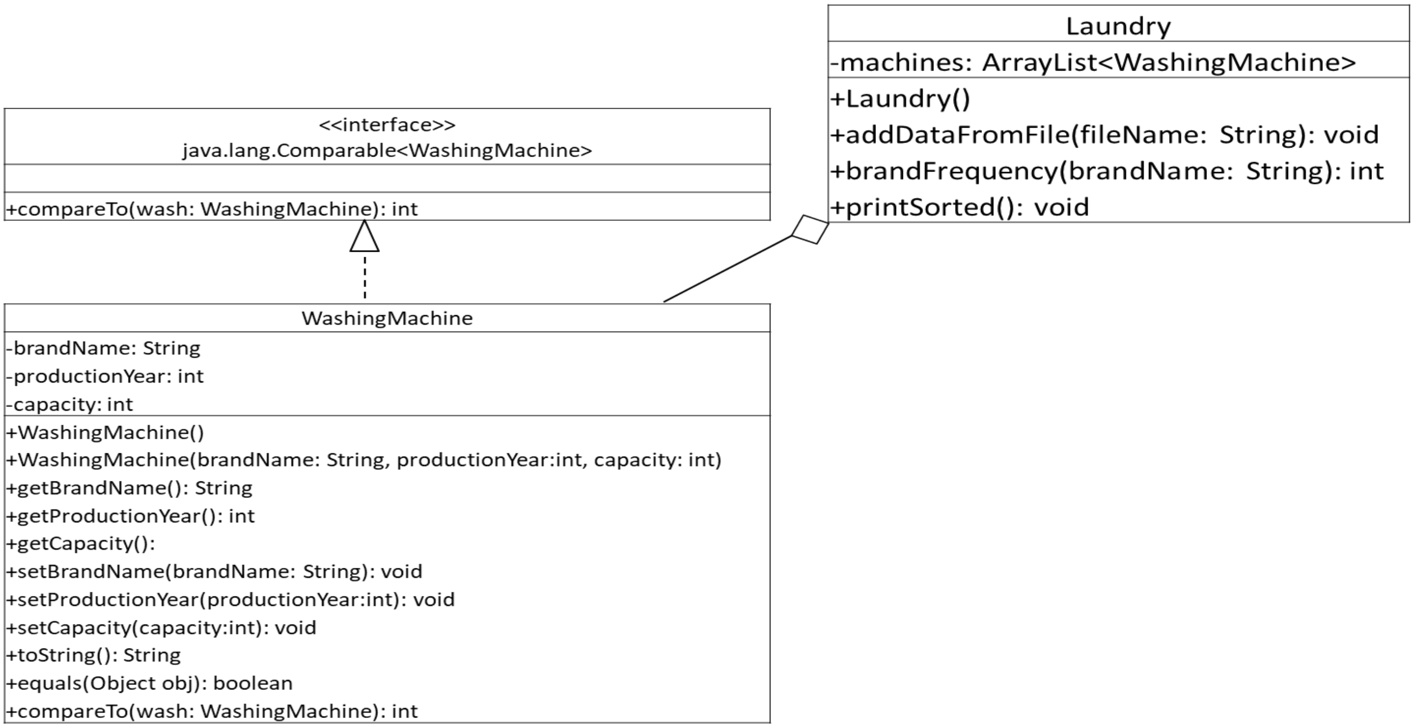
```
//11 marks as in comments
//1
import java.io.*;
import java.util.*;
//1
public class Test {
  public static void main(String[] args) throws FileNotFoundException {
    //1
    ArrayList<WashingMachine> machines = new ArrayList<>();
    //3 for reading data from file
    Scanner in = new Scanner(new File("washing.txt"));
    String bn;
    int py, c;
    WashingMachine wm;
    while (in.hasNext()) {
      bn = in.next();
      py = in.nextInt();
      c = in.nextInt();
      wm = new WashingMachine(bn, py, c);
      if (machines.contains(wm))
        System.out.println("error: this machine has been added before");
      else
        machines.add(wm);
    }
    in.close();
    //3 for computing and printing
    int max = -1;
    wm = null;
    for (WashingMachine w:machines)
      if (w.getCapacity() > max) {
        max = w.getCapacity();
        wm = w;
      }
    System.out.println("Machine with max capacity:\n" + wm);
    //2 for sorting and printing
    Collections.sort(machines);
    System.out.println("Machines after sorting:");
    for (WashingMachine w:machines)
      System.out.println(w);
  }
}
```

Q4:

```
//1 mark
error: this machine has been added before
Machine with max capacity:
brandName= Samsung, productionYear= 2021, capacity= 10
Machines after sorting:
brandName= Beko, productionYear= 2022, capacity= 8
brandName= Bosch, productionYear= 2023, capacity= 9
brandName= Samsung, productionYear= 2021, capacity= 10
brandName= Toshiba, productionYear= 2022, capacity= 6
brandName= Wansa, productionYear= 2020, capacity= 9
```

Sample 2 from the Questions Bank

Consider the following class diagram. Answer the given questions using **Object Oriented Programming** style.


Laundry
-machines: ArrayList<WashingMachine>
+Laundry()
+addDataFromFile(fileName: String): void
+brandFrequency(brandName: String): int
+printSorted(): void
<<interface>>
java.lang.Comparable<WashingMachine>
+compareTo(wash: WashingMachine): int
WashingMachine
-brandName: String
-productionYear: int
-capacity: int
+WashingMachine()
+WashingMachine(brandName: String, productionYear:int, capacity: int)
+getBrandName(): String
+getProductionYear(): int
+getCapacity():
+setBrandName(brandName: String): void
+setProductionYear(productionYear:int): void
+setCapacity(capacity:int): void
+toString(): String
+equals(Object obj): boolean
+compareTo(wash: WashingMachine): int

Question 1: [10 marks]

Write the code of class **WashingMachine**. The comparison in **compareTo()** method is based on **productionYear** .

Question 2: [13 marks]

a) Create a text file includes the following data of 6 **WashingMachine** objects, where the values represent brandName, productionYear and capacity, respectively:

Toshiba 2022 6
Samsung 2021 10
Bosch 2023 9
Beko 2022 8
Bosch 2020 9
Beko 2022 8

b) Write the code of class **Laundry** if you know that:

- The constructor of the class creates a new empty ArrayList and assign it to the **machines** instance variable.

- The method **addDataFromFile()** takes as an argument the name of a text file, and read data from the file and store it in the **machines** list. It is not allowed to add the same **WashingMachine** twice and in this case an error message should be displayed.
- The method **brandFrequency**() takes as an argument **brandName**, and returns number of **WashingMachine** objects in the **machines** list of the same **brandName**.
- The method **printSorted()** sorts the **machines** list according to **productionYear**, then print its elements on the screen (one object per line).

Question 3: [6 marks]

Write a Test class to test classes declared in Q1 and Q2 as follows:

1. Declare and create a **Laundry** object referenced by a variable called **dry**.
2. Fill the **machines** list of the object **dry** by data from the text file created in Q2 by calling a suitable method.
3. Print number of **WashingMachine** objects of **brandName**: **Bosch** in the object **dry** by calling a suitable method.
4. Sort then print the **WashingMachine** objects in the object **dry** by calling a suitable method.

Question 4: [1 mark]

Copy and paste the exact output of your code.

End of Questions

Answers:

Q1:

```
//10 marks as in comments
//1
public class WashingMachine implements Comparable<WashingMachine> {
  //1
  private String brandName;
  private int productionYear;
  private int capacity;
  //1
  public WashingMachine() { }
  //1
  public WashingMachine(String brandName, int productionYear, int capacity) {
    this.brandName = brandName;
    this.productionYear = productionYear;
    this.capacity = capacity;
  }
  //2
  public String getBrandName() { return brandName; }
  public void setBrandName(String brandName) {this.brandName = brandName; }
  public int getProductionYear() { return productionYear; }
  public void setProductionYear(int productionYear) {
    this.productionYear = productionYear;
  }
  public int getCapacity() { return capacity; }
  public void setCapacity(int capacity) { this.capacity = capacity; }
  //1
  @Override
  public String toString() {
    return "brandName= " + brandName + ", productionYear= " + productionYear +
            ", capacity= " + capacity;
  }
  //1
  @Override
  public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof WashingMachine)) return false;
    WashingMachine that = (WashingMachine) o;
    if (productionYear != that.productionYear) return false;
    if (capacity != that.capacity) return false;
    return brandName.equals(that.brandName);
  }
  //2
  @Override
  public int compareTo(WashingMachine wash) {
    return productionYear - wash.productionYear;
  }
}
```

Q2:

```
//13 as in comments
//1
import java.io.*;
import java.util.*;
//1
public class Laundry {
  private ArrayList<WashingMachine> machines;
  //1
  public Laundry() {
    machines = new ArrayList<>();
  }
  //4: 1 for the header and 3 for the body
  public void addDataFromFile(String fileName) throws FileNotFoundException {
    Scanner in = new Scanner(new File(fileName));
    String bn;
    int py, c;
    WashingMachine wm;
    while (in.hasNext()) {
      bn = in.next();
      py = in.nextInt();
      c = in.nextInt();
      wm = new WashingMachine(bn, py, c);
      if (machines.contains(wm))
        System.out.println("error: this machine has been added before");
      else
        machines.add(wm);
    }
    in.close();
  }
  //3: 1 for the header and 2 for the body
  public int brandFrequency(String brandName) {
    int count = 0;
    for (WashingMachine w:machines)
      if (w.getBrandName().equals(brandName))
        count++;
    return count;
  }
  //3: 1 for the header and 2 for the body
  public void printSorted() {
    Collections.sort(machines);
    for (WashingMachine w: machines)
      System.out.println(w);
  }
}
```

Q3:

```
//6 marks as in comments
//1
import java.io.FileNotFoundException;
//1
public class Test {
  public static void main(String[] args) throws FileNotFoundException {
    //4: 1 for each line
    Laundry dry = new Laundry();
    dry.addDataFromFile("washing_machines.txt");
    System.out.println("Number of Bosch machines: " + dry.brandFrequency("Bosch"));
    System.out.println("Machines after sorting:");
    dry.printSorted();
  }
}
```

Q4:

```
//1 mark
error: this machine has been added before
Number of Bosch machines: 2
Machines after sorting:
brandName= Bosch, productionYear= 2020, capacity= 9
brandName= Samsung, productionYear= 2021, capacity= 10
brandName= Toshiba, productionYear= 2022, capacity= 6
brandName= Beko, productionYear= 2022, capacity= 8
brandName= Bosch, productionYear= 2023, capacity= 9
```