CPS 181 – Exam 1 review of chapter 1:

The goal of this exam review is to help you see how everything in the world can be thought of as an object or a class. You will design your own classes in Java and practice using constructors, methods, setters/getters, static fields, and ArrayLists.

**Instructions:**

Pick a real-life object (Student, Dog, Car, PizzaOrder). Follow each part below to gradually build your class. Remember:   
- Nouns (things it has) become fields (variables).  
- Verbs (things it can do) become methods.

**Part 1: Classes & Objects**

Define a class with at least 2 fields and 1 method.

Example:

*class Student {  
 String name;  
 int id;  
  
 void study() {  
 System.out.println(name + " is studying.");  
 }  
}*  
  
Student s1 = new Student();

**Part 2: Constructors**

Add a constructor to your class that sets the fields when you create an object.

Example:

*public Student(String n, int i) {  
 name = n;  
 id = i;  
}*

**Part 3: Constructor Overloading**

Add a second constructor with a different parameter list to give another way of creating your object.

Example:

*public Student(String n) {  
 name = n;  
 id = -1;  
}*

**Part 4: Mutators (Setters) & Accessors (Getters)**

Add at least one setter and one getter to your class.

Example:

*public void setName(String n) {  
 name = n;  
}  
  
public int getId() {  
 return id;  
}*

**Part 5: Static Fields**

Add a static field to track how many objects have been created. Update it in the constructor.

Example:

*static int totalStudents = 0;  
  
public Student(String n, int i) {  
 name = n;  
 id = i;  
 totalStudents++;  
}*

**Part 6: ArrayList of Objects**

In your main method, create an ArrayList of your class and add at least 3 objects to it.

Example:

*ArrayList<Student> students = new ArrayList<>();  
students.add(new Student("Alice", 1));  
students.add(new Student("Bob", 2));  
students.add(new Student("Charlie", 3));*

My example was a hair salon.

Code below:

package chapter1Exam;

import java.util.ArrayList;

// Part 1: Classes & Objects. Create a class named HairSalon with fields & methods.

class HairSalon{

String name;

String address;

String openingHours;

int phoneNumber;

// Method

void displayDetails(){

System.***out***.println("Salon Name: " + name);

System.***out***.println("Address: " + address);

System.***out***.println("Opening Hours: " + openingHours);

System.***out***.println("Phone Number: " + phoneNumber);

}

// Part 5: Static Fields.

// Add a static field to your class to keep track of the number of HairSalon objects created.

// Update this/increment this field in the constructors.

static int *salonCount* = 0;

// Part 2: Constructors. Add a constructor to the HairSalon class to initialize fields.

public HairSalon(String n, String a, String hours, int phone){

name = n;

address = a;

openingHours = hours;

phoneNumber = phone;

*salonCount*++; // P5. Increment the static field.

}

// Part 3: Constructors Overloading.

// Add a second constructor with a different parameter list to give another way of creating your object.

public HairSalon(String n, String a){

name = n;

address = a;

openingHours = "10 AM - 6 PM"; // default hours

phoneNumber = 987654321; // default phone number

*salonCount*++; // P5. Increment the static field.

}

// Optional method to display the salon count

public static void displaySalonCount() {

System.***out***.println("Total salons created: " + *salonCount*);

}

// Part 4: Mutators (Setters) & Accessors (Getters).

// Add at least one setter and one getter method to your class.

public void setOpeningHours(String hours){

openingHours = hours;

}

public String getOpeningHours(String hours){

return openingHours;

}

public void setName(String n){

name = n;

}

public String getName(String n){

return name;

}

public void setAddress(String a){

address = a;

}

public String getAddress(String a){

return address;

}

public void setPhoneNumber(int phone){

phoneNumber = phone;

}

public int getPhoneNumber(int phone){

return phoneNumber;

}

public static void main(String[] args) {

/\*\*

Part 1 code:

this part is commented out to avoid redundancy since the object creation and field assignment

is already done in the main method below.

// Create an object of HairSalon

HairSalon salon = new HairSalon();

// Assign values to fields

salon.name = "Elegant Cuts";

salon.address = "123 Main St, Cityville";

salon.openingHours = "9 AM - 7 PM";

salon.phoneNumber = 1234567890;

// Call method to display details

HairSalon.displayDetails();

\*/

// After part 2,3,4,5 code:

/\*

HairSalon s1 = new HairSalon("Elegant Cuts", "123 Main St, Cityville", "9 AM - 7 PM", 1329846593);

HairSalon s2 = new HairSalon("Quick Trim", "456 Oak Rd");

s1.displayDetails();

s2.displayDetails();

// Show how many salons have been created

HairSalon.displaySalonCount(); // prints: Total Salons created: 2

\*/

//Part 6: Create an ArrayList of your class and add at least 3 objects to it.

ArrayList<HairSalon> hairsalon = new ArrayList<>();

hairsalon.add(new HairSalon("Elegant Cuts", "123 Main St, Cityville", "9 AM - 7 PM", 1329846593));

hairsalon.add(new HairSalon("Quick Trim", "456 Oak Rd"));

hairsalon.add(new HairSalon("Style Studio", "789 Pine Ave", "10 AM - 8 PM", 1987654321));

// Display details of each salon in the ArrayList

for(HairSalon salon : hairsalon){

salon.displayDetails();

System.***out***.println(); // for better readability

}

}

}

Output:

Salon Name: Elegant Cuts

Address: 123 Main St, Cityville

Opening Hours: 9 AM - 7 PM

Phone Number: 1329846593

Salon Name: Quick Trim

Address: 456 Oak Rd

Opening Hours: 10 AM - 6 PM

Phone Number: 987654321

Salon Name: Style Studio

Address: 789 Pine Ave

Opening Hours: 10 AM - 8 PM

Phone Number: 1987654321