

```
import java.awt.*;

public class Person {
    private String name;
    private int age;
    private int weight;
    private Color hairColor;

    public Person() {
        name = "Untitled";
        age = 21;
        weight = 150;
        hairColor = Color.BLACK;
    }

    public Person(String n, int a, int w, Color h) {
        name = n;
        age = a;
        weight = w;
        hairColor = h;
    }

    public void setName(String s) { name = s; }
    public void setAge(int a) { age = a; }
    public void setWeight(int w) { weight = w; }
    public void setHairColor(Color c) { hairColor = c; }

    public String getName() { return name; }
    public int getAge() { return age; }
    public int getWeight() { return weight; }
    public Color getHairColor() { return hairColor; }

    public String toString() {
        return name + ", " + age + ", " + weight;
    }

    public boolean equals(Object o) {
        Person other = (Person)o;
        return name.equals(other.name);
    }
}
```

Person Questions

- 1) What two words are really after “public class Person” but aren’t written there?
- 2) Do the Java classes Object and Person demonstrate Composition or Inheritance?
- 3) Do the Java classes Person and String demonstrate Composition or Inheritance?
- 4) How many constructors does the Person class have?
- 5) Write the copy constructor for the Person class.

Answer the following questions about these lines of code in a Driver file:

```
Person p1 = new Person();
Person p2 = new Person("Bob", 40, 160, Color.BROWN);
Person p3 = new Person("Bob", 60, 180, Color.BLACK);
Person p4 = null;
```

- 6) What is printed by this statement:

```
System.out.println(p1.toString()); //System.out.println(p1); does the same thing!
```

- 7) Circle the if-cases that evaluate to true. 8) The Person class _____ the Object class’ equals(Object) and toString() methods

```
if(p2 == p2) {
}
if(p2 == p3) {
}
if(p2.equals(p2)) {
}
```

- 9) What happens if this line of code is executed?

```
p4.setName("Bob");
```

```

import java.awt.*;

public class Student extends Person {
    private int id;

    public Student() {
        super();          //explicitly calls Person() constructor
        id = 0;
    }

    public Student(int i) {
        //super() is here even though I didn't write it!
        id = i;
    }

    public Student(String n, int a, int w, Color h, int i) {
        super(n, a, w, h);    //calls different Person() constructor
        id = i;
    }

    public int getID() { return id; }
    public void setID(int i) { id = i; }

    public String toString() {
        return super.toString() + ", " + id;    //calls super class version + id
    }
}

```

```

import java.awt.*;

public class Teacher extends Person {
    private int salary;

    public Teacher() {
        super();
        salary = 0;
    }

    public Teacher(String n, int a, int w, Color h, int s) {
        super(n, a, w, h);
        salary = s;
    }

    public int getSalary() { return salary; }
    public void setSalary(int s) { salary = s; }

    public String toString() {
        return super.toString() + ", $" + salary;
    }
}

```

In a Driver file somewhere...

10) Which of these lines of code are okay?

```
Person p = new Person();
Student s = new Student();
Teacher t = new Teacher("Bob", 40, 160, Color.RED, 70000);
Teacher t2 = new Teacher("A", 0, 0, Color.RED);
```

11) Which of these lines of code are okay?

```
int age = p.getAge();
int age2 = s.getAge();
int salary = p.getSalary();
int salary2 = s.getSalary();
```

12) What does the following code print? Is it the same as System.out.println(s)?

```
System.out.println(s.toString());
```

13) Assume this method is added to the Teacher class. Why won't this compile? How would you fix it?

```
//returns the product of the Teacher's age and salary
public int ageSalary() {
    return age * salary;
}
```

14) Assume that the Student class is modified so that it overrides the setAge() method. What is the purpose of this override? Why do you think the word super is used? What would happen if super was not used? Does this effect the Person class?

```
public void setAge(int a) {
    if(a < 0)
        super.setAge(-a);
    else
        super.setAge(a);
}
```

15) What is the first line inside this constructor? (I'm trying to be tricky...)

```
public Fraction() {
    numer = 0;
    denomer = 1;
}
```