1. Best loop to implement a sentinel loop, a counting loop, and a y/n loop?

for loop is best for counting loop. while loop or do-while loop are best for sentinel loop and y/n loop.

2. Set up a method to receive some data and perform a certain task:

a. A method that accepts 3 values and returns smallest

public int smallest(int x, int y, int z) {

int smaller = x;

if (y < smaller)

smaller = y;

if (z < smaller)

smaller = z;

return smaller;

}

b. A method that accepts an array and returns the average

public double average(int[] array) {

double sum = 0.0;

for (int i = 0; i < array.length; ++i) {

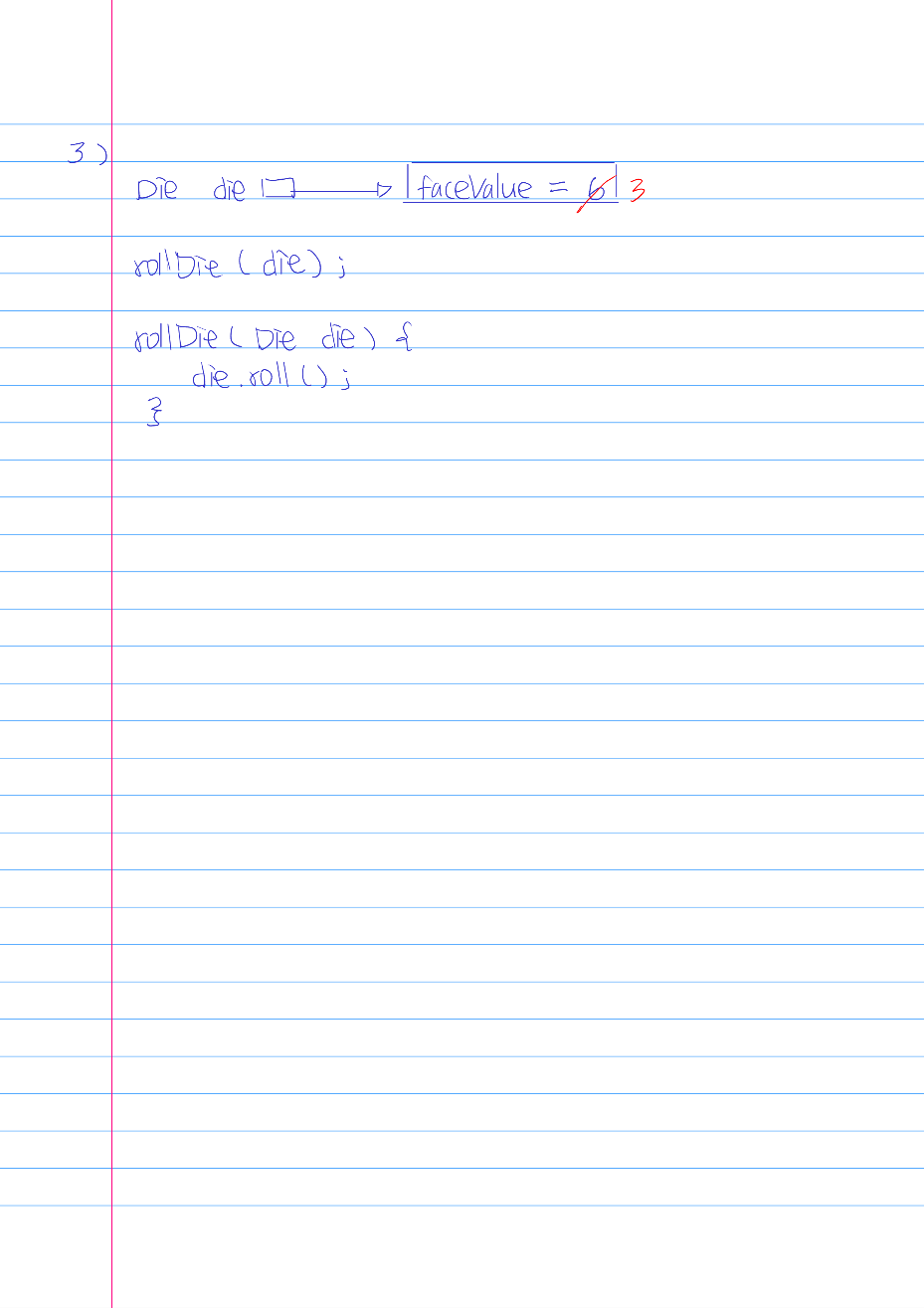
sum += array[i];

}

Return sum / array.length;

}

3. An object such as Die object is sent to a method as a parameter and it was modified inside the method, what happened to the original object? Draw a diagram showing those variables.



4. Differences between black box and white box testing.

Two approaches of software testing.

Black box testing is a software testing done by people without having knowledge of internal logic or code structure. Black box testing is focused on input and expected output of the software application.

White box testing is a software testing done by developers. Developers focus on the internal logic and code structure. In white box testing, code is visible to the testers.

5. Differentiate testing from debugging.

Testing attempts to ensure that the program will solve the intended problem under all the constraints specified in the requirements.

Debugging is the process of determining the cause of a problem and fixing it.

6. What are some levels of testing from earliest to latest?

Unit testing

Integration testing

System testing

Acceptance testing

Regression testing

7. Steps to debug a program.

Reproduce the error.

Track down the error.

Isolate the error.

Fix the error.

8. Perform a simple if-else with the conditional operator.

Boolean head = (coin.isHeads() ? true : false);

9. Determine max of three strings or three int values.

public Comparable max(Comparable val1, Comparable val2, Comparable val3) {

Comparable maxVal = val1;

if (maxVal.compareTo(val2) < 0) {

maxVal = val2;

}

if (maxVal.compareTo(val3) < 0) {

maxVal = val3;

}

Return maxVal;

}

10. Given a, b, and c, arrange them from smallest to largest.

if (a <= b && a <= c) {

System.out.println(a);

if (b < c) {

System.out.println(b);

System.out.println(c

}

else {

System.out.println(c);

System.out.println(b);

}

}

else if (b <= a && b <= c) {

System.out.println(b);

if (a < c) {

System.out.println(a);

System.out.println(c);

}

else {

System.out.println(c);

System.out.println(a);

}

}

else {

System.out.println(c);

if (a < b) {

System.out.println(a);

System.out.println(b);

}

else {

System.out.println(b);

System.out.println(a);

}

}

11. Use a nested loop to generate a pattern like a rectangle or a triangle.

##### or \*

##### \*\*

##### \*\*\*

Rectangle:

for (int width = 0; width < 3; ++width) {

for (int length = 0; length < 5; ++length) {

System.out.print(“#”);

}

System.out.println();

}

Triangle:

for (int height = 0; height < 3; ++height) {

for (int base = 0; base < height + 1; ++base) {

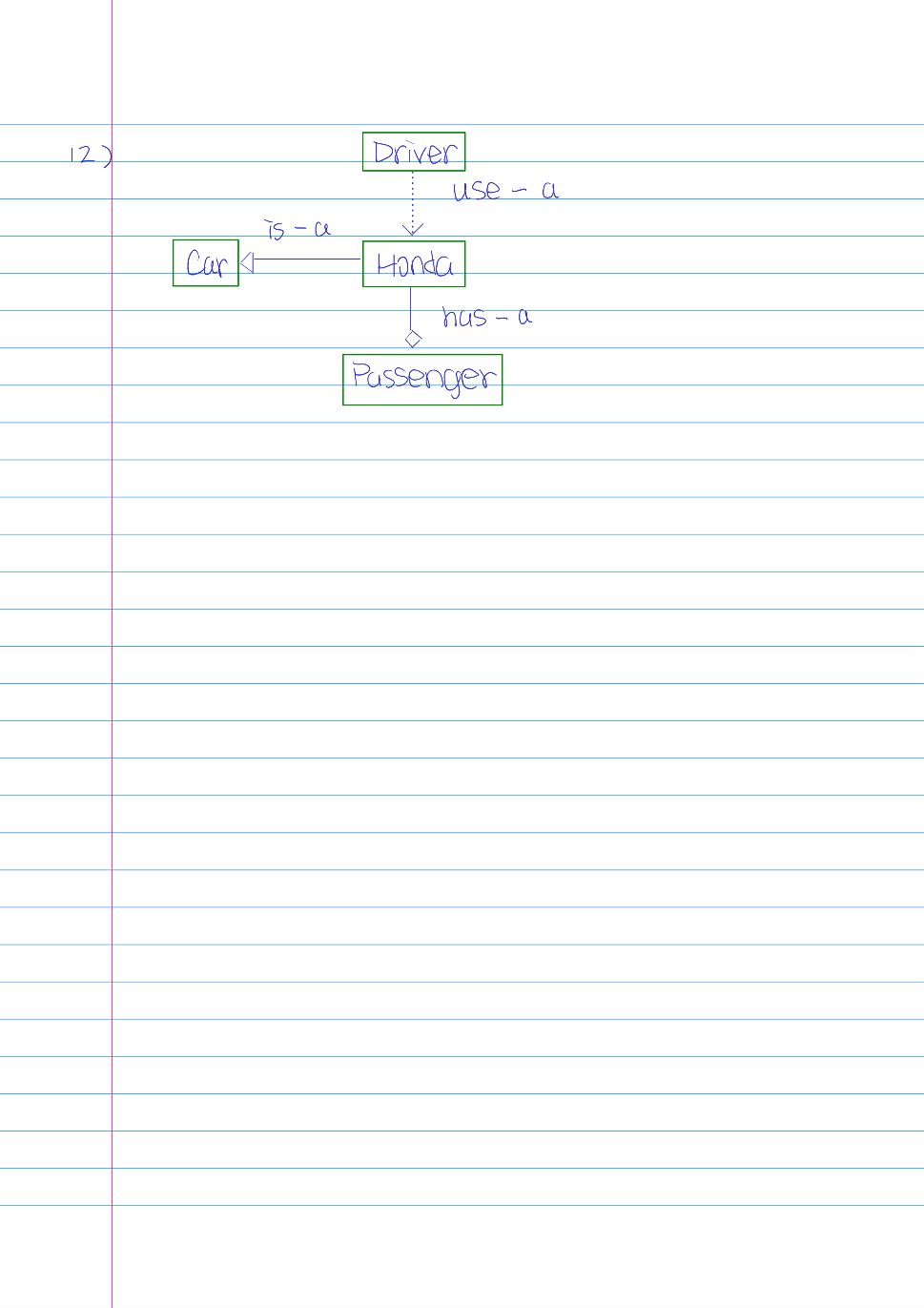
System.out.print(“\*”);

}

System.out.println()

}

12. Draw a simple UML class diagram showing relationships between some classes.



13. Use the Coin or Die class to perform a simulation:

a. Toss 2 coins 100 times and count the number of times they are the same.

Coin coin1 = new Coin();

Coin coin2 = new Coin();

int same = 0;

for (int i = 0; i < 100; ++i) {

coin1.flip();

coin2.flip();

if (coin1.isHeads() && coin2.isHeads() || !coin1.isHeads() && !coin2.isHeads()) {

++same;

}

}

b. Toss 2 dice 100 times and count the number of times they are the same.

Die die1 = new Die();

Die die2 = new Die();

int same = 0;

for (int i = 0; i < 100; ++i) {

if (die1.roll() == die2.roll()) {

++same;

}

}

14. Use an array to keep track of the counts when rolling two dice 1000 times.

Die die1 = new Die();

Die die2 = new Die();

int[] rollResults = new int[12];

for (int i = 0; i < 1000; ++i) {

++rollResults[die1.roll() + die2.roll – 1];

}

15. How is a static variable different from an instance variable?

A static variable associates with the class rather than with an object of that class. A static variable can only be referenced by static methods.

16. How is a static method different from a non-static method

A static method in a class can be invoked without instantiated an object of that class.