1. We did questions like this in Unit 1, but we always need more practice knowing how data changes when passed into methods. What is printed?

In Dog.java:

public class Dog {

private int tricks; //number of tricks the dog knows

private String name; //the dog's name

public Dog(String newName) {

tricks = 0;

name = newName;

}

public int getTricks() { return tricks;}

public String getName() { return name;}

public void addTrick() { tricks++;}

public void train(int n){ tricks += n;}

public String toString(){ return name + " knows " + tricks + " tricks.";}

}

In Main.java:

public class Main {

public static void main(String[] args) {

Dog fido, spot, rover;

int x, y, z;

fido = new Dog("Fido");

spot = new Dog("Spot");

x = 6;

y = 8;

fido.addTrick();

*school1*(fido,x);

*school2*(spot,x);

rover = *nextGeneration*(fido,y);

y = *mystery*(x);

System.*out*.println(fido + "\n" + spot + "\n" + rover + "\nx = " + x + "\ny = " + y);

}

public static void school1(Dog dog, int n) { dog.train(n); }

public static void school2(Dog dog, int n) {

n = 4;

dog = new Dog("Rover");

dog.train(n);

}

public static Dog nextGeneration(Dog origDog, int x) {

String name = origDog.getName() + " II";

int numTricks = origDog.getTricks() + x;

Dog nextDog = new Dog(name);

nextDog.train(numTricks);

return nextDog;

}

public static int mystery(int x) {

int y = 5 + x;

x = y/3;

return x;

}

}

1. In this question I want you to practice the difference between instanced and static fields. What is printed?

In FieldPractice.java:

public class FieldPractice {

private static int x = 7;

private int y;

public FieldPractice() {

y = 3;

}

public void doStuff(FieldPractice otherFP){

FieldPractice.x++;

this.y--;

FieldPractice.x++;

otherFP.y++;

}

public int getY(){

return this.y;

}

public static int getX(){

return FieldPractice.x;

}

}

In Main.java:

class Main {

public static void main(String[] args) {

FieldPractice fp1 = new FieldPractice();

FieldPractice fp2 = new FieldPractice();

fp1.doStuff(fp2);

fp2.doStuff(fp2);

fp2.doStuff(fp1);

fp1.doStuff(fp2);

System.out.println(FieldPractice.getX());

System.out.println(fp1.getY());

System.out.println(fp2.getY());

}

}

**Use these classes for the next few problems:**

In class HockeyPlayer.java:

public class HockeyPlayer {

public HockeyPlayer(){}//If there are no instance fields, you can actually just skip writing a constructor, but I have it here so that you can see what it would automatically do.

public void play(){

System.out.println("I am playing hockey!");

}

}

In class Goalie.java:

public class Goalie extends HockeyPlayer {

public Goalie(){}

public void play(){

super.play();

System.out.println("You won't get the puck past me!");

}

}

In class Defenseman.java:

public class Defenseman extends HockeyPlayer {

public Defenseman(){}

public void play(){

super.play();

System.out.println("I am going to stop the other forwards!");

super.play();

}

}

In class Forward.java:

public class Forward extends HockeyPlayer {

public Forward(){}

public void play(){

System.out.println("I am going to score a goal!");

super.play();

}

}

3. This question allows you to practice subclasses and overridding. What is printed?

In class Main.java:

class Main {

public static void main(String[] args) {

Goalie g = new Goalie();

Defenseman d = new Defenseman();

Forward f = new Forward();

f.play();

d.play();

g.play();

}

}

4. This question is practice in polymorphism. What is printed?

In class Main.java:

class Main {

public static void main(String[] args) {

HockeyPlayer h = new HockeyPlayer();

HockeyPlayer h2 = new Goalie();

h.play();

h2.play();

}

}

5. This question is also practice in polymorphism. Which lines of code would cause errors? Bonus - give yourself a pat on the back if you can correctly identify if they are compile or runtime errors!

In class Main.java:

class Main {

public static void main(String[] args) {

Forward f1 = new Forward();

Goalie g1 = new Goalie();

HockeyPlayer h1 = new Forward();

HockeyPlayer h2 = g1;

Goalie g2 = (Goalie) h2;

Goalie g3 = (Goalie) h1;

Goalie g4 = (Goalie) f1;

g1.save();

h2.save();

g2.save();

}

}