**Name**

**Advanced Programming in Java**

**Lab Exercise 9/24/2019**

Reference: Lesson 15 in Blue Pelican Java

1. double length = 44.0;

int width =13;

Rectangle myRect = new Rectangle(length, width);

a. Identify the class

b. Identify the object

c. What type of parameter(s) are passed to the constructor?

2. Write out the signature for the constructor of the *Rectangle* class from #1 above.

3. Suppose a constructor for the *Lunch* class is as follows:

public Lunch(boolean diet, int cal)

{

diet\_yes\_no = diet;

calories = cal;

}

Write appropriate code that will create a *Lunch* object called *yummy5*. You should

tell the constructor that, yes, you are on a diet, and the number of calories should

be 900.

4. BankAccount account39 = new BankAccount(500.43);

a. Identify the class

b. Identify the object

c. What type of parameter(s) are passed to the constructor?

5. A class is like a \_\_\_\_\_\_\_\_\_\_\_. An object is like a \_\_\_\_\_\_\_\_\_\_\_\_.

Fill in the blanks above using the word “cookie” and “cookie cutter”.

6. What’s wrong (if anything) with the following constructor for the *School* class?

public void school(int d, String m)

{ … some code … }

7. Which of the following is a correct association?

a. One class, many objects

b. One object, many classes

8. Which must exist first?

a. The class

b. The object

9. Is the following legal? If not, why?

//Constructor

//This code is in *main* of *Tester* class

public House(int j, boolean k)

int p = 3, q = 9;

{ …some code… }

House myHouse = new House(p, q);

10. //Constructor

public Band(int numMembers, int numInstruments, String director, double amount)

{ …code…}

Band ourBnd = new Band(mem, instrmnts, “Mr. Perkins”, budget);

What should be the data types of:

**a.** mem

**b.** instrmnts

**c.** budget

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class ActionStory

{

public int var1;

public double var2;

public String sss;

public void Flintstone(double zorro)

{ …some code…}

public String getDino( )

{ …some code…}

public ActionStory(String x, int y, double z)

{ …some code… }

}

11. From the *ActionStory* class above, write the signature of the constructor.

12. From the *ActionStory* class above, what is/are the instance field(s).

13. From the *ActionStory* class above, write the signature(s) of the all the method(s).

14. Write code that instantiates an object called *terminator* from the *ActionStory* class. Pass the following parameters to the constructor:

The integer should be 19, the *String* should be “Ralph”, and the *double* should be

24.18.

15. Assume an object called *gravy* has been created from the *ActionStory* class. Write code that will set the state variable *var2* to 106.9 for the *gravy* object.

16. Write code that will print the value of the *ActionStory* data member, *sss*. Assume you

have already created an object called *bart*.

17. Again, assume we have an object called *bart* instantiated from the *ActionStory* class.

What should you fill in for **<#1>** below in order that *sss* be stored in the variable *jj*?

**<#1>** jj = bart.sss;

18. Create a class called *Trail*. It should have instance fields *x* and *y* that are integers.

Instance field *s* should be a *String*. The constructor should receive a *String* which is

used to initialize *s*. The constructor should automatically set *x* and *y* both equal to 10.

There should be a method called *met* that returns a *String* that is the hex equivalent of

*x\* y*. This method receives no parameters.

19. Suppose you wish to call a method whose signature is:

*public double peachyDandy(int z)*

Write code that would call this method (assume we have an object name *zippo*). Also

assume that this code will be placed in the *main* method of a *Tester* class and that the

*peachyDandy* method is in some other class.

20. Refer to the information in 19 above. What’s wrong with trying to call this method in

the following fashion?

double hamburger = zippo.peachyDandy(127.31);

.

**Project… What’s That Diameter?**

Create a new method for the *Circle* class called *diameter*. Add this method to the *Circle* class below. It should return a *double* that is the diameter of the circle. No parameters are passed to this method.

// Circle.java

public class Circle

{

public double radius; //This is a **State Variable**…also called **Instance**

//**Field** and **Data Member.** It is available to code

// in ALL the methods in this class.

//This part is called the **constructor** and lets us specify the radius of a

//particular circle.

public Circle(double r)

{

radius = r;

}

//This is a **method**. It performs some action (in this case it calculates the

//area of the circle and returns it.

public double area( ) //**area method**

{

double a = Math.PI \* radius \* radius;

return a;

}

public double circumference( ) //**circumference method**

{

double c = 2 \* Math.PI \* radius;

return c;

}

}

In a *Tester* class, test the performance of your new *diameter* method as follows:

(Your project should have two classes, *Tester* and *Circle.*)

public class Tester

{

public static void main( String args[] )

{

Circle cir1 = new Circle(35.5);

System.out.println( cir1.diameter( ) );

}

}

The printout should give 71.0 as the answer.

**Project … Overdrawn at the Bank**

Create a class called *BankAccount*. It should have the following properties:

1. Two state variables:

*balance*…*double*… This is how much money is currently in the account.

*name*…*String*…The name of the person owning the account.

2. Constructor should accept two parameters.

a. One should be a *double* variable that is used to initialize the state variable,

*balance*.

b. The other should be a *String* that is used to initialize the state variable, *name*.

3. Two methods:

a. *deposit*…returns nothing…accepts a *double* that is the amount of money being

deposited. It is added to the *balance* to produce a new balance.

b. *withdraw*…returns nothing…accepts a *double* that is the amount of money

being taken out of the account. It is subtracted from the *balance* to produce a

new *balance*.

Create a *Tester* class that has a *main( )* method. In that method you should input from the

keyboard the amount (1000) of money initially to be put into the account (via the constructor)

along with the name of the person to whom the account belongs.

1. Use these two pieces of data to create a new *BankAccount* object called *myAccount*.

2. Call the *deposit* method to deposit $505.22.

3. Print the *balance* state variable.

4. Call the *withdraw* method to withdraw $100.

5. Print the remaining *balance* in this form:

The Sally Jones account balance is, $1405.22