

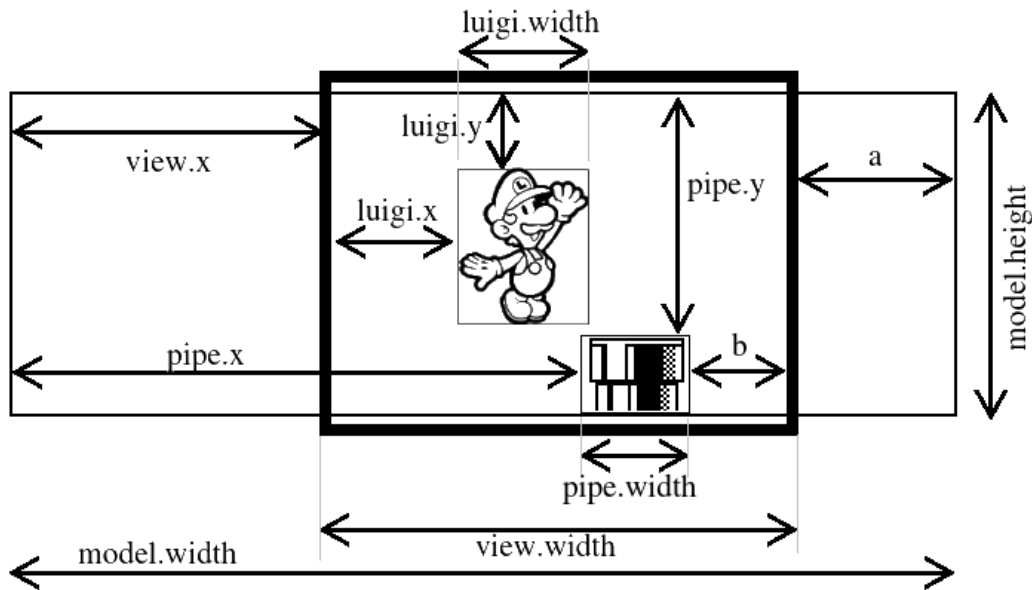
1. (2 points) What is your name?
2. (5 points) Add a member variable to the Graph class (in the right column of this page) to hold a collection of Vertex objects. Also add a member variable to hold a collection of Edge objects.
3. (5 points) Add a constructor to Graph class, and initialize the member variables in this constructor.
4. (5 points) Add a method to the Graph class with the signature,
void addVertex(int x, int y)
Implement this method to add a new Vertex to the collection of vertices with the specified coordinates.
5. (5 points) Add a method named “addEdge” that lets the user add an edge to the graph.
6. (5 points) Add a method named “edgeCount”, that returns the number of edges in the graph. Also, check to make sure the number of edges is not less than the number of vertices. If it is, throw an informative RuntimeException.
7. (6 points) Add a copy constructor to the Edge class that makes a deep copy (not a shallow copy) of the Edge.
8. (5 points) In Java, when a primitive type is passed as a parameter,
 - a new object is allocated on the heap
 - its value is copied on the stack.
 - its memory location is copied on the stack..
 - the memory location of the object enclosing the primitive type is discovered by the garbage collector and is connected with the receiving function.

```
class Vertex
{
    int x;
    int y;
}

class Edge
{
    Vertex a;
    Vertex b;
}

class Graph
{

```



9. (5 points) Consider the game represented in the diagram above. (Note that the programmer used view coordinates to represent Luigi's position, and model coordinates to represent the Pipe's position, which is not very consistent.)

Give an expression for the length 'a'.

10. (5 points) Give an expression for 'b'.

11. (4 points) Suppose you are paused in a debugger, and observe the following values:

```

luigi.height = 43
luigi.width = 12
luigi.x = 50
luigi.y = 50
pipe.height = 200
pipe.width = 20
pipe.x = 267
pipe.y = 60
model.height = 200
model.width = 800
view.width = 150
view.x = 33

```

At this moment, is Luigi colliding with the pipe?

12. (7 points) (Continuing from the previous question...) Why? Circle ONE answer.

- $33 - 20 > 12$
- $50 + 43 > 60$
- $267 > 33 + 50 + 12$
- $50 + 43 < 200$
- $200 - 60 > 50 + 43$
- $50 + 33 < 267$

13. (5 points) Suppose a class named Toyota extends a class named Car. Which of these lines will cause a compile error?

- `Toyota t = new Car();`
- `Car c = new Toyota();`

14. (5 points) In Java, if you pass a variable of type "int" as a parameter to some function, it

- Implicitly calls the integer constructor.
- Passes the value in a member variable.
- Passes the reference in a member variable.
- Puts the memory location of the variable on the stack.
- Puts the value on the stack.
- Puts the memory location of the variable on the heap.

15. (7 points) Which will cause a NullPointerException to occur?

- Run your code in a debugger.
- Pass a a value of null as a parameter.
- Set a variable to null when the programmer expects it to be non-null.
- Attempt access a member variable using a null reference.
- Set a member variable to null after it had a non-null value.
- Test whether a variable is equal to null.
- Modify a collection while an iterator is iterating over it.
- Dividing by zero.

16. (5 points) Observe how this code uses local variables in the constructor:

```
class Game {
    Avatar av;
    Background bg;
    Game() {
        Avatar av = new Avatar();
        Background bg = null;
    }
    void onClick(int x, int y) {
        if(x < 0 || y < 0)
            throw new
                RuntimeException("!");
        int i;
        av.teleport(x, y);
        if(bg == null)
            System.out.println(i);
    }
}
```

When “onClick” is called, a NullPointerException will be thrown. How would you fix it? (Circle one)

- Initialize bg with “new Background()”.
- Pass valid values for x and y.
- Remove “Avatar ” and “Background ” from the constructor.
- Initialize “i” with “int i = 0;”.
- change the “==” to “!=”.
- Pass an Avatar object and a Background object as parameters to the constructor.
- Declare Avatar to be “static”.
- Declare Background to be “static”.
- Remove the “throw” line from this code.

17. (5 points) ConcurrentModificationException is thrown when

- You modify a variable that already references an object.
- You modify a variable that is equal to null.
- Generic types are used improperly.
- You call Iterator.remove to remove a non-existent object.
- An iterator detects that the contents of its collection have changed.
- You modify an object to which multiple variables refer.

18. (6 points) Consider this nonsensical code:

```
class Pizza extends Object // 1
{
    static Toppings t = new Toppings(); // 2
    Sauce s; // 2
    void slurp(BreadStick b) // 3
    {
        if(this.s == null) // 4
            this.b = null; // 5
        else
            t = null; // 6
    }
    static Sauce swallow() // 7
    {
        if(t == null) // 8
            return this.s; // 9
        else
            return null; // 10
    }
}
```

Of the 10 commented lines, which two contain errors that the compiler will catch?

19. (8 points) In the following code, **use overriding** to implement an isTube method that returns true if (and only if) the Sprite instance used to call the method is a Tube.

```
class Sprite
{
}

class Tube extends Sprite
{
}
```