Encapsulation Name Kaito Hikino

Encapsulation **means hiding the details of a class by making the data private and forcing client programmers to use public "get" and "set" methods (called** *accessors* **and** *mutators***) to access and modify the data.**

1. Fill in the blanks with **public** or **private**:

```
class Widget
{
private double dNum;

private String sWord;

public Widget()
{
dNum = 2.5;
sWord = "Hello";
}
```

double sum()

{

double dTemp = dNum + sWord.length();

return dTemp;

}

//and lots more java

2. Circle the one variable that is NOT a data member, and should not be declared as **public** or **private.**

3. Write *accessor* and *mutator* methods for the **double** member variable:

public void getNum()

{

return dNum;

}

public void setNum(int num_)

{

num_ = dNum ;

}

4. Write *accessor* and *mutator* methods for the **String** member variable

```
public void getWord()
{
return sWord;
}
```

```
public void setWord(String word_)
{
word_ = sWord;
}
```

5. Create an *instance* of the **Widget** class named Bob. Use the **set** methods to set Bob's **dNum** to 5.5 and Bob's **sWord** to be "Hi!".

```
Widget bob;
public void setup()
{
bob = new Widget();
bob.setNum(5.5);
bob.setWord("Hi!");
}
```

6. Create a second *instance* of the **Widget** class named Sue. Use the **get** and **set** methods to set Sue's member variables to the same values as Bob's.

```
Widget sue;
public void setup()
{
sue = new Widget();
sue.setNum(5.5);
sue.setWord("Hi!");
```

```
}
```