CS 2113 - Java 2 & 3: OOP

1. Mark which of the **bold** lines below would be allowed (check mark) and which would be forbidden (big X) by Java’s encapsulation system.

```
public class X {
  public    int a;
  private   int b;
  protected int c;
  public static int d;

  private   int func1() {…}
  protected int func2() {…}
  public    int func3() {…}
}

public class Y extends X{
  public int m;
  public void func4() {
    m = func1();
    m = func2();
    m = a;
    m = b;
    m = c;
    m = d;
  }

}

// Assume this code is in the main function in class Z
int var;
var = X.a;
var = X.b;
var = X.c;
var = X.d;
var = X.m;

X x1 = new X();
var = x1.a;                x1.func1();
var = x1.b;                x1.func2();
var = x1.c;                x1.func3();
var = x1.d;                x1.func4();
var = x1.m;

Y y1 = new Y();
var = y1.a;                y1.func1();
var = y1.b;                y1.func2();
var = y1.c;                y1.func3();
var = y1.d;                y1.func4();
var = y1.m;
```

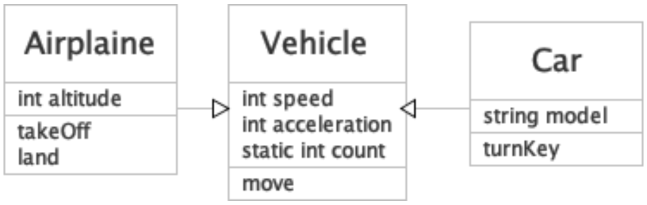
2. Design a UML Class Diagram to represent how you might structure this system.

A voice mail system records calls to multiple mailboxes. The system records each message, the caller's number, and the time of the call. The owner of a mailbox can play back saved messages and see a list of the messages they have received.

CS 2113 - Java 2 & 3: OOP

3. Fill in the data members and method prototypes for this UML diagram.

```
public class Vehicle
```



```
 public Vehicle(){
   count++;
 }
```

Stack

Address	*Name*	*Contents*
10000	s	65
10008	v	
10016	v2	
10016	c	

```
public class Airplane
```

Heap

Address	*Contents*

```
public class Car
```

```
public static void main() {
  int s = 65;
  Vehicle v;
  Vehicle v2 = new Vehicle();
  v2.speed = s;
  Car c = new Car("Honda");
}
```

CS 2113 - Java 2 & 3: OOP

Self-Quiz

1. What are static variables and methods in Java? Why does it make sense that the main function is always static?

2. What are the different visibility settings for variables in a Java class? Why would you use each one?

3. How do you create an array of objects? What do you need to remember to do? How does this relate to C programming and memory?