**APCS Test on Classes & Test Classes 2011-12 Name\_Ethan Turkeltaub\_\_\_\_\_\_\_\_\_\_\_\_\_ /190 Points**

**Complete all parts as best you can. You may use Word or notepad but NOT blueJ or any such IDE. Please leave time to print your final work!!**

**Part I. Short Answer:** **/20 points**

**List the three parts of a Java Class and provide a concise explanation of each IN YOUR OWN WORDS. Include an explanation of a default constructor.**

**1. Fields: Fields are variables, and it is what the Object “knows”. They are defined at the top of the class with public String name; and then a value is assigned to them in the constructor or later in the class.**

**2. Constructor(s): A constructor is how an object is created from a class. The default constructor (with no parameters) is what is the default knowledge or values of fields. If you have a constructor with parameters, you can define fields as what you’d like.**

**3. Method(s): Defined as accessors or mutators, a method allows you to communicate with the Object. With accessors, you can get information. With mutators, you can change information.**

**Part II.**

Create a class to represent a **SchoolBus**. Include fields, a constructor, and methods as follows:

**SchoolBus** objects should store the following information: **/20 points**

* + **numSeats (int), numPassengers (int), busCompany (String), mileage (double), busNumber (int), driver (String),** and **hasSeatBelts(boolean).**

There are TWO ways to create a new **SchoolBus**  **/30 points**

* 1. The default SchoolBus has 60 seats, starts with 0 passengers, is owned by M & J company, starts with 0.0 miles, has a bus number of 101, is driven by Joe Jensen and does NOT have seat belts.
  2. The number of seats, bus company, bus number, driver, and if the bus has seat belts or not are all input via parameter variables. The number of passengers always starts at 0 and the mileage always starts at 0.0

A **SchoolBus** has the following **methods**:

* + **getNumSeats** – returns the number of seats **/5 points**
  + **getNumPassengers** – returns the number of passengers **/5 points**
  + **getCompany** – returns the bus company **/5 points**
  + **getMileage** – returns the current mileage. **/5 points**
  + **getBusNumber** – returns the bus number **/5 points**
  + **getDriver**– returns name of the current bus driver **/5 points**
  + **getHasSeatBelts** – returns the state of the boolean field **hasSeatBelts** **/5 points**
  + **printInfo** – prints current value of all fields in a format that is easy to understand; *part of this method is done for you.* **/15 points**
  + **changeDriver** – changes the name of current driver to the name of a new driver given by parameter  
     **/10 points**
  + **drive –** increases mileage by an amount given by parameter  
     **/10 points**
  + **pickupPassengers** **–** increases the number of passengers by a number given by parameter  
     **/10 points**

public class SchoolBus {

// Fields

private int numSeats;

private int numPassengers;

private String busCompany;

private double mileage;

private int busNumber;

private String driver;

private boolean hasSeatBelts;

private int seatsAvailable;

// Constructors

public SchoolBus() {

numSeats = 60;

seatsAvailable = 60;

numPassengers = 0;

busCompany = "M & J";

mileage = 0.0;

busNumber = 101;

driver = "Joe Jensen";

hasSeatBelts = false;

}

public SchoolBus(int inSeats, String inCompany, int inNumber, String inDriver, boolean inSeatBelts) {

numSeats = inSeats;

seatsAvailable = inSeats;

numPaseengers = 0;

busCompany = inCompany;

mileage = 0.0;

busNumber = inNumber;

driver = inDriver;

hasSeatBelts = inSeatBelts;

}

// Methods

public int getNumSeats() {

return numSeats;

}

public String getCompany() {

return busCompany;

}

public double getMileage() {

return mileage;

}

public int getBusNumber() {

return busNumber;

}

public String getDriver() {

return driver;

}

public boolean getHasSeatBelts() {

return hasSeatBelts;

}

public int getSeatsAvailable() {

return seatsAvailable;

}

public void printInfo() {

System.out.println("Information for " + busCompany + " bus #" + busNumber + ":");

System.out.println("Passengers: " + numPassengers);

System.out.println("Seats: " + numSeats);

System.out.println("Current Driver: " + driver);

System.out.println("Seatbelts: " + hasSeatBelts);

System.out.println("Mileage: " + mileage);

}

public void changeDriver(String newDriver) {

driver = newDriver;

System.out.println("Driver changed to " + newDriver);

}

public void drive(double miles) {

mileage = mileage + miles;

System.out.println("The bus now has " + mileage + " miles");

}

public void pickupPassengers(int newPassengers) {

if (newPassengers > seatsAvailable) {

System.out.println("Sorry, not enough seats available!")

}

else {

numPassengers = numPassengers + newPassengers;

seatsAvailable = seatsAvailable - newPassengers;

System.out.println("There are currently " + seatsAvailable + " seats left after that pickup.");

}

}

}

**Part III:**  Complete the test class TestSchoolBus to create 2 SchoolBus objects correctly and test their methods as specified in the comments supplied below. **/30 points**

public class TestSchoolBus {

public static void main(String[] args) {

SchoolBus b1 = new SchoolBus();

SchoolBus b2 = new SchoolBus(120, "Blue Jay", 42, "Joe Schmoe", true);

b1.printInfo();

b2.printInfo();

b1.changeDriver("Dan Dawson");

b2.pickupPassengers(5);

b1.drive(9.0);

b1.printInfo();

b2.printInfo();

System.out.println(b2.getNumber());

}

}

**Part IV:**  List all accessors and mutators for the SchoolBus class. **/10 points**

**Accessors:** getNumSeats(), getCompany(), getMileage(), getBusNumber(), getDriver(), getHasSeatBelts(), getSeatsAvailable(), printInfo()

**Mutators:** changeDriver(String newDriver), drive(double miles), pickupPassengers(int newPassengers)