

[COURSES](#) > [FALL 2007 INTRO PROGRAMMING.COMP110.004](#) > [CONTROL PANEL](#) > [ASSIGNMENTS](#) > [PREVIEW ASSESSMENT HOMEWORK 5](#)

Preview Assessment Homework 5

Name: Homework 5

Instructions: Your assignment must be properly submitted on BlackBoard by the deadline for it to be graded.

Completing your submission:

- When submitting, be sure to press the "Submit" button.
- The Save button does NOT submit your assignment.

Verify that your submission is complete

- In BlackBoard, select the "My Grades" section in the main menu
- The icon next to the assignment should be a an exclamation mark (!).

Having trouble with BlackBoard:

- Try using a different computer to submit your homework on BlackBoard.
- You might be blocked out of the assignment and cannot access it. Also in the "My Grades" section a lock icon appears in place of your assignment. Email Dorian and Will and we will reset your entry.

Multiple Attempts: This Test allows multiple attempts.

Force Completion: This Test can be saved and resumed later.

▼ Question Completion Status:

Question 1

10 points

Save

Below is the Automobile class.

```
public class Automobile {
    // data members
    private String name;
    private double milesPerGallon;
    private double gasInTank; // measured in gallons
    public static final int SPEEDLIMIT = 55;

    public Automobile(String name, double mpg, double startTankGal){
        this.name = name;
        this.milesPerGallon = mpg;
        this.gasInTank = startTankGal;
    }

    private double gasUsed(double miles){
        return miles/milesPerGallon;
    }

    // The automobile drives X miles and the tank is reduced by Y gallons
    public void drive(double miles){
        gasInTank = gasInTank - gasUsed(miles);
    }

    // Display the information on the automobile
    public String toString(){
```

```

return name + "\t remaining tank " + gasInTank + "\t (" + milesPerGallon +)";
}

// getter/ setter methods
public String getName(){
return name;
}

public void setName(String name){
this.name = name;
}

public double getMilesPerGallon(){
return milesPerGallon;
}
}

```

Indicate whether each of the statements is valid or invalid for compilation within a main method. Assume the main method is in a separate class from the Automobile class. Note that some of the valid statements cannot stand alone in a main method. The question demonstrates the meaning of the Java class keywords, public vs private and static. Here is the start of the main method.

```

public static void main(String[] args) {

Automobile car = new Automobile("Car", 30, 15);
// consider statements used after this point
}

```

- | | | |
|--------------------------|-------------------------|------------|
| <input type="checkbox"/> | car.drive(50) | A. Valid |
| <input type="checkbox"/> | car.gasUsed(20) | B. Invalid |
| <input type="checkbox"/> | Automobile.drive(20) | |
| <input type="checkbox"/> | Automobile.gasUsed(20) | |
| <input type="checkbox"/> | car.getName() | |
| <input type="checkbox"/> | car.setName("Race car") | |
| <input type="checkbox"/> | car.name | |
| <input type="checkbox"/> | getName() | |
| <input type="checkbox"/> | Automobile.SPEEDLIMIT | |
| <input type="checkbox"/> | car.SPEEDLIMIT | |

Question 2**10 points**[Save](#)

Use the given Automobile class and write a main method that produces the following output. The question demonstrates creating class instances of Automobile, using methods on the objects, and displaying the values of the Automobile objects. The answer should only be the contents within the main method.

There are two Automobiles, a hybrid and SUV with 50 and 20 miles per gallon respectively.

Both automobiles start with 15 gallons of gas.

Output:

Drive 100 miles
 Hybrid remaining tank 13.0 (50.0)
 SUV remaining tank 10.0 (20.0)
 Drive 100 more miles (200 miles total)
 Hybrid remaining tank 11.0 (50.0)
 SUV remaining tank 5.0 (20.0)



Question 3

20 points

Save

Complete the code skeleton of the Employee class (given below); that is fill-in the missing code. The Employee class stores an employee's name and wage per hour. The data should be private to encapsulate the implementation. The methods include getter and setter methods to access the class data members. The toString method displays the Employee data as specified in the comments. The weeklyPay method computes the weekly pay: hours less than 40 hours (OVERTIME) are paid at the wage level. Extra hours past 40 hours (OVERTIME) are paid at 1.5 times the wage. Your answer should include the entire skeleton with the missing code inserted. Using the Employee class you write, the following main method should produce the following output.

Main method:

```
public static void main(String[] args) {

    // Create instance of employee
    Employee jack = new Employee("Jack", 9.50);

    // calculate pay
    double pay = 0.0;

    pay = jack.weeklyPay(Employee.OVERTIME);
    System.out.println(jack + ", pay = " + pay + " for hours = " + Employee.OVERTIME);

    pay = jack.weeklyPay(Employee.OVERTIME + 5);
    System.out.println(jack + ", pay = " + pay + " for hours = " + (Employee.OVERTIME+5));

    // Raise wage and calculate pay
```

```
jack.setWage(12.00);
pay = jack.weeklyPay(15);
System.out.println(jack + ", pay = " + pay + " for hours = " + 15);
}
```

Output

```
(Jack, $9.5/hour), pay = 380.0 for hours = 40
(Jack, $9.5/hour), pay = 451.25 for hours = 45
(Jack, $12.0/hour), pay = 180.0 for hours = 15
```

Employee code skeleton

```
public class Employee {

    // declare data members (must specify public/private/static)

    // constant
    public static final int OVERTIME = 40;

    // constructor
    public Employee(String n, double w){

    }

    // getter/setters used to access private data members
    public String getName(){

    }

    public void setName(String n){

    }

    public double getWage(){

    }

    public void setWage(double w){

    }

    // Create a string display the name and wage with this format
    // (Name, $wage/hour)
    // example name = Jack, wage = 9.5
    // (Jack, $9.5/hour)
    public String toString(){

    }

    public double weeklyPay(int hours){
```

}

}

Normal 3 Times New Roman **B** *I* U [List Bulleted] [List Numbered] [List Indented] [List Nested] [List Nested] [List Nested]

Save Submit