AP Computer Science A   Test 2   Name:________________

Fall 2016   Mr. Alwin Tareen

Part I. (35 points) Solve each of the following problems. For the multiple choice problems, select the correct answer by placing an "X" in the box beside it.

(1pt) **1.** Which of the following choices are considered fundamental qualities of good object-oriented design?

□ catch and release
□ bait and switch
□ state and behaviour
□ divide and conquer

1 pt

(1pt) **2.** Which of the following choices is the correct way to set up a constructor?

□ `public String Rectangle()`
□ `public static int SportsTeam(int players)`
□ `public constructor Bicycle(int gears)`
□ `public Compass(int direction)`

1 pt

(1pt) **3.** Consider a class that has two constructors. Which of the following conditions must be true in order for the program to compile correctly?

□ The constructors must be declared private and void.
□ The constructors must be placed in separate source code files.
□ The constructors must specify a return type.
□ The constructors must have unique parameter lists.

1 pt

(1pt) **4.** Which of the following choices would be considered an accessor method?

□ `public Kennel(double price)`
□ `public String getName()`
□ `public Ticket()`
□ `public class Player`

1 pt

(1pt) **5.** Which of the following is a fundamental quality of mutator methods?

□ Mutator methods always return an integer data type.
□ The instance variables are declared within mutator methods.
□ Mutator methods alter the instance variables.
□ Mutator methods change all the data types of the class.

1 pt

(1pt) **6.** Which of the following correctly describes the purpose of the `toString()` method?

□ It gives client programs the ability to easily display the instance variables of a class.
□ It constructs an object, and allocates sufficient memory for it.
□ It alters the instance variables of a class.
□ It assigns the correct data type to each of the class' variables.

1 pt

(1pt) **7.** The following is a statement in a `Java` program which compiles and executes correctly.

`submarine.dive(depth);`

Which of the following choices can be inferred from the above statement?

□ `dive` must be the name of an instance variable.
□ `dive` must be a method.
□ `submarine` must be the name of a class.
□ `submarine` must be a method.

1 pt

7 pts

Consider the following implementation of the `Student` class:

```
public class Student
{
    // instance variables
    private String name;
    private double sum;
    private int numGrades;

    // constructors
    public Student(String n)
    {
        <CODE>
    }

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

    public double getAverage()
    {
        return sum/numGrades;
    }

    // mutator methods
    public void setGrade(int grade)
    {
        sum += grade;
        numGrades++;
    }
}
```

(1$^{\text{pt}}$) **8.** Which of the following should replace `<CODE>` such that the instance variable `name` is correctly initialized when a new object is created?

□ `String name = n;`

□ `name = n;`

□ `n name;`

□ Cannot be done because `name` is `private`.

1 pt

(1$^{\text{pt}}$) **9.** Assuming that `<CODE>` is filled in correctly, how would you create a `Student` object called `pupil` and set `name` to `"Sally"`?

□ `Student pupil = new Student();`

□ `pupil.name = "Sally";`

□ `pupil = new Student("Sally");`

□ `Student pupil = new Student("Sally");`

1 pt

2 pts

(1^{pt}) **10.** Which of the following would print the `name` of the student represented by the object called `bart`?

1 pt

□ `System.out.println(bart.getName());`

□ `System.out.println(bart.name());`

□ `System.out.println(bart(name));`

□ `System.out.println(name(bart))`

(1^{pt}) **11.** Assume a `Student` object called `lisa` has been created and grades have been assigned. How would you correctly retrieve this student's average?

1 pt

□ `int average = lisa.getAverage();`

□ `double average = getAverage(lisa);`

□ `double average = lisa.getAverage();`

□ `lisa.setGrade(98);`

(2^{pts}) **12.** Consider the following `Java` source code for `PassingParameters.java`:

2 pts

```java
public class PassingParameters
{
    public static void displayTotal(int total)
    {
        total = 75;
        System.out.println(total);
    }

    public static void main(String[] args)
    {
        int score = 10;
        displayTotal(score);
        System.out.println(score);
    }
}
```

What will be the output when this program is executed? Write your answer in the box below:

The Terminal Display Output of `PassingParameters.java`

4 pts

(4pts) **13.** Consider the following incomplete implementation of the `Rectangle` class:

4 pts

```java
public class Rectangle
{
    // instance variables
    private int length;
    private int width;

    // constructors
    public Rectangle(int len, int wid)
    {
        length = len;
        width = wid;
    }
}
```

(a) (2 pts) Write an accessor method called `getPerimeter()` which calculates and returns the perimeter of the rectangle.

(b) (2 pts) Write an accessor method called `getArea()` which calculates and returns the area of the rectangle.

`APLine` Question(2010 AP CompSci Free Response)

(9pts) **14.** An `APLine` is a line defined by the equation $ax + by + c = 0$, where a is not equal to zero, b is not equal to zero, and a, b, and c are all integers. The slope of an `APLine` is defined to be the `double` value $-a/b$. A point(represented by integers x and y) is on an `APLine` if the equation of the `APLine` is satisfied when those x and y values are substituted into the equation. That is, a point represented by x and y is on the line if $ax + by + c$ is equal to 0. Examples of two `APLine` equations are shown in the following table.

9 pts

Equation	Slope($-a/b$)	Is point $(5, 2)$ on the line?
$5x + 4y - 17 = 0$	$-5/4 = -1.25$	Yes, because $5(5) + 4(-2) + (-17) = 0$
$-25x + 40y + 30 = 0$	$25/40 = 0.625$	No, because $-25(5) + 40(-2) + 30 \neq 0$

Assume that the following code segment appears in a class other than `APLine`. The code segment shows an example of using the `APLine` class to represent the two equations shown in the table.

```
APLine line1 = new APLine(5, 4, -17);
double slope1 = line1.getSlope();     // slope1 is assigned -1.25
boolean onLine1 = line1.isOnLine(5, -2); // true

APLine line2 = new APLine(-25, 40, 30);
double slope2 = line2.getSlope();     // slope2 is assigned 0.625
boolean onLine2 = line2.isOnLine(5, -2); // false
```

Write the `APLine` class. Your class must produce the indicated results when invoked by the code segment given above. You may ignore any issues related to integer overflow. Your implementation must include:

(a) (1 pt) The declaration of the private instance variables `a`, `b` and `c`.

(b) (2 pts) A constructor that has three integer parameters that represent `a`, `b`, and `c`, in that order. You may assume that the values of the parameters representing `a` and `b` are not zero.

(c) (3 pts) A method `getSlope()` that calculates and returns the slope of the line.

(d) (3 pts) A method `isOnLine(int x, int y)` that returns `true` if the point represented by its two parameters (`x` and `y`, in that order) is on the `APLine`, and returns `false` otherwise.

Write your solution on the next page.

9 pts

Complete `APLine.java` in the space below.

0 pts

The Restaurant Question

(9pts) **15.** After graduation, you have decided to open your own restaurant to earn some money for college. Since real estate in Beijing is expensive, your restaurant can only accommodate 8 customers at a time. You have decided to put your coding skills to good use by designing a Java program to keep track of your customers' meals, bills and discounts. You have chosen to design a Customer class and a Restaurant class to accomplish this task.

9 pts

The class Customer has been provided for you. It defines the instance variables meal and price, which are associated with each customer. It also defines the accessor and mutator methods for each of these instance variables.

Java Source Code for Customer.java

```java
public class Customer
{
    // instance variables
    private String meal;
    private double price;

    // constructors
    public Customer(String m, double p)
    {
        meal = m;
        price = p;
    }

    // accessor methods
    public String getMeal()
    {
        return meal;
    }

    public double getPrice()
    {
        return price;
    }

    // mutator methods
    public void setMeal(String m)
    {
        meal = m;
    }

    public void setPrice(double p)
    {
        price = p;
    }
}
```

0 pts

Write the `Restaurant` class. It must include an `array` data structure that will contain each of the `Customer` objects in your restaurant. Also, your class must produce the indicated results when invoked by the test bench given below. *Note:* Assume that the `toString()` method has been written for you.

Your `Restaurant` implementation must include:

(a) (1 pt) The declaration of the private instance variable `patron`, which is an array of type `Customer`.

(b) (2 pts) A constructor with no parameters, which initializes the array `patron` to be of size `8`.

(c) (2 pts) A mutator method called `addCustomer(int i, Customer c)` which inserts a `Customer` object into the `patron` array at the specified index `i`.

(d) (4 pts) A mutator method called `applyDiscount()` which applies a 25% discount to each of your customers' bills. *Hint:* beware of `null` objects in the `patron` array!

Java Source Code for `RestaurantTest.java`, the Test Bench

```java
public class RestaurantTest
{
    public static void main(String[] args)
    {
        Customer charlie = new Customer("Burger", 10.0);
        Customer dennis = new Customer("Salad", 8.0);

        Restaurant grillery = new Restaurant();
        grillery.addCustomer(0, charlie);
        grillery.addCustomer(1, dennis);

        System.out.println(grillery);
        grillery.applyDiscount();
        System.out.println(grillery);
    }
}
```

The Terminal Display Output of `RestaurantTest.java`

```
Burger $10.0
Salad $8.0

Burger $7.5
Salad $6.0
```

Write your solution on the next page.

9 pts

Complete `Restaurant.java` in the space below.

0 pts