AP Computer Science
Fall 2014

Test 2

Name: _____
Mr. A. Tareen

Part I. (44 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 is the specific term for the type of variables that are declared at the beginning of a class?

☐ `intrinsic` ☐ `instance` ☐ `incognito` ☐ `inverted`

1 pt

(1pt) **2.** The *default constructor* of a class has what type of data in its parameter list?

☐ integers ☐ doubles ☐ nothing ☐ Strings

1 pt

(1pt) **3.** How would you identify the constructor of a class?

☐ It must be declared with the keyword `constructor`.
☐ It always returns an `integer`.
☐ It has the same name as the class.
☐ It must use the return type `void`.

1 pt

(1pt) **4.** A method that accesses a class object without altering that object is called:

☐ An absolute method.
☐ A retrieval method.
☐ An adverse method.
☐ An accessor method.

1 pt

(1pt) **5.** A method that changes the state of an object by modifying at least one of its instance variables is called:

☐ A mutator method.
☐ A mixer method.
☐ A motivator method.
☐ A masher method.

1 pt

(1pt) **6.** When two or more methods in the same class have the same name but different parameter lists, they are called:

☐ Extreme methods.
☐ Overloaded methods.
☐ Massive methods.
☐ Major methods.

1 pt

(1pt) **7.** Which of the following statements could be used to create a new object called `b1`, from a class named `BankAccount`?

☐ `b1 = BankAccount();`
☐ `BankAccount b1 = new BankAccount();`
☐ `b1(BankAccount) = new BankAccount();`
☐ `BankAccount.b1 = new BankAccount();`

1 pt

7 pts

(1pt) **8.** Which of the following is *not* part of a Java class?

- ☐ A constructor.
- ☐ Instance variables.
- ☐ Accessor methods.
- ☐ A method to delete the object and release its memory.

1 pt

(1pt) **9.** Which invokes the method `length()` on the object `str` and stores the result in `val`?

- ☐ `val = str.length();`
- ☐ `val = length.str();`
- ☐ `val = length().str;`
- ☐ `val = length(str);`

1 pt

Questions 10-13 refer to the following `Date` class declaration:

```
public class Date
{
    private int myDay;
    private int myMonth;
    private int myYear;

    public Date()                           // default constructor
    {
        ...
    }

    public Date(int mo, int day, int yr)    // constructor
    {
        ...
    }

    public int month()                      // returns month of Date
    {
        ...
    }

    public int day()                        // returns day of Date
    {
        ...
    }

    public int year()                       // returns year of Date
    {
        ...
    }

    // Returns String representation of Date as "m/d/y", e.g. 4/18/1985
    public String toString()
    {
        ...
    }
}
```

2 pts

(3pts) **10.** Which of the following correctly constructs a `Date` object?

3 pts

- ☐ `Date d = new (2, 13, 1947);`
- ☐ `Date d = Date(2, 13, 1947);`
- ☐ `Date d; d = new (2, 13, 1947);`
- ☐ `Date d; d = Date(2, 13, 1947);`
- ☐ `Date d = new Date(2, 13, 1947);`

(3pts) **11.** Which of the following will cause an error message?

3 pts

I
```
Date d1 = new Date(8, 2, 1947);
Date d2 = d1;
```

II
```
Date d1 = null;
Date d2 = d1;
```

III
```
Date d = null;
int x = d.year();
```

- ☐ I only
- ☐ II only
- ☐ III only
- ☐ II and III only
- ☐ I, II and III

(3pts) **12.** A client program creates a `Date` object as follows: `Date d = new Date(1, 13, 2002);` Which of the following subsequent code segments will cause an error?

3 pts

- ☐ `String s = d.toString();`
- ☐ `int x = d.day();`
- ☐ `Date e = d;`
- ☐ `Date e = new Date(1, 13, 2002);`
- ☐ `int y = d.myYear;`

(3pts) **13.** A method in a client program for the `Date` class has this declaration:

3 pts

```
Date d1 = new Date(month, day, year);
```

where `month`, `day` and `year` are previously defined integer variables. The same method now creates a second `Date` object `d2` that is an exact copy of the object `d1` refers to. Which of the following code segments will *not* do this correctly?

I
```
Date d2 = d1;
```

II
```
Date d2 = new Date(month, day, year);
```

III
```
Date d2 = new Date(d1.month(), d1.day(), d1.year());
```

- ☐ I only
- ☐ II only
- ☐ III only
- ☐ II and III only
- ☐ All will do this correctly.

12 pts

(10pts) **14.** The three most important numbers describing a telescope are: the `diameter` of the main lens(the one in front), the focal length of the main lens(`mainLength`), and the focal length of the eyepiece(`eyeLength`). Each of these numbers should be considered a `double`. Write the `Telescope` class. Your implementation should include the following:

10 pts

(a) (1 pt) The declaration of the private instance variables `diameter`, `mainLength` and `eyeLength`.

(b) (3 pts) A constructor that has three variables in its parameter list. These will initialize the values of the private instance variables `diameter`, `mainLength` and `eyeLength`.

(c) (3 pts) An accessor method called `magnification()`, which calculates and returns the magnification of the telescope in the form of a `double`. It is calculated with the formula: magnification = `mainLength/eyeLength`

(d) (3 pts) An accessor method called `fNumber()`, which calculates the focal number of the telescope, and returns it as a `double`. It is calculated with the following formula: fNumber = `mainLength/diameter`

10 pts

(3pts) **15.** For the following Java program, write the output in the box labelled `Terminal Output`.

3 pts

```
public class PassByValue
{
    public static int mystery(int x, int y)
    {
        return x + y;
    }

    public static void main(String[] args)
    {
        int a = 7;
        int b = 2;
        int result = 0;
        result = mystery(a, b);
        System.out.println("Result is: " + result);
    }
}
```

`Terminal Output`

(10pts) **16.** 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.

10 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
```

3 pts

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) (3 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.

10 pts