Java Name –
Classes Objects AP Style Worksheet #1 Period –

Multiple Choice – Circle your answer on this paper AND fill in the Scantron form.

1. As we have studied so far, a client program has _____ while a regular class such as `Bug` usually doesn't have _____ .
a. a `main` method
b. a default constructor
c. an IDE
d. at least one `String` variable

2.
```
public class Bug
{
   private String mySpecies;
   private int myAge;
   private double myWeight;
   private Position myPosition;     // position (x, y) in pond
   private int myHealth;            // 0 (dead) to 100 (healthy)
   . . .
```

Given the partial `Bug` class above, which method is the **best candidate** for being a static method?
a. `move` `// bug moves to a new grid location`
b. `eat` `// bug eats and gains weight`
c. `getWeight` `// returns weight of bug`
d. `getGridSize` `// returns size of the grid where the bug lives`

3. Which of the following statements about objects is **true**?
a. An object usually has private methods
b. Only one object variable can be instantiated in a client class
c. Every object belongs to a class.
d. The interface of a class is hidden from the user.
e. The name of an object should begin with the prefix `my`

4. Which of the following statements about a constructor is **false**?
a. a default constructor may accept parameters
b. a constructor does not return a value
c. at most, there can only be one default constructor in a class
d. it is possible to have more than one constructor with parameters in a class

5. Another term for a property is
a. an object variable b. a class c. an instance field d. an instantiator e. an object reference

6. An object stores its state in
a. reference variables b. properties c. `double` variables d. a constructor e. a `String`

7. Consider the following three method headers.
I. `double doSomething(String a, int b)`
II. `int doSomething (String name, int num)`
III. `double doSomething (int b, String a)`

Which of the following statements is true about their method signatures?
a. I and II have the same method signature and this signature is different from the method signature of III.
b. II and III have the same method signature and this signature is different from the method signature of I.
c. I and III have the same method signature and this signature is different from the method signature of II.
d. I, II, and III all have the same method signature.
e. I, II, and III all have different method signatures.

8. Suppose a `Car` class has the following properties.

```
private double myMileage;     // mileage on odometer
private double myMPG;         // miles per gallon
private double myTankSize;    // size of gas tank in gallons
```

What is true about the initialization of the properties?
a. The properties must be initialized in the constructor. If they are not, there will be a compile-time error.
b. The properties must be initialized in the constructor. If not, the program will compile but there will be a run-time error when the `Car` object constructor is called.
c. The properties should be initialized in the constructor. If not, they will automatically be initialized to `null`.
d. The properties should be initialized in the constructor. If not, they will be automatically be initialized to 0.

9. If two methods have the same name and the same signatures, then….
a. it is an example of overloading.
b. an error occurs.
c. aliasing occurs.
d. there are two valid copies of the same method.

10. Which of the following statements about objects and classes is **not** true?
a. Object references are stored in object variables.
b. If it is not initialized in a client program, an object variable is automatically set equal to `null`.
c. An object reference is the memory address of an object.
d. Each object variable stores its own unique memory address.
e. A class allows a programmer to instantiate objects.

For the following exercises, consider the `BankAccount` class whose incomplete definition is shown below.

```
// A bank account has a balance that can be changed by deposits
public class BankAccount
{
	public BankAccount(double initialBalance)
	{
		myBalance = initialBalance;
	}
	public void deposit(double amount)
	{
		double newBalance = myBalance + amount;
		myBalance = newBalance;
	}
	public double getBalance()
	{
		return myBalance;
	}
	private double myBalance;
}
```

11. Suppose the following statements are executed in a client program:

```
BankAccount b1 = new BankAccount(500);
BankAccount b2 = new BankAccount();
b1.deposit(b2.getBalance());
b2.deposit(b1.getBalance());
```

What are the balances of `b1` and `b2` after the code is executed?
a. `b1` has balance = 500, `b2` has balance = 500
b. `b1` has balance = 1000, `b2` has balance = 500
c. `b1` has balance = 500, `b2` has balance = 1000
d. `b1` has balance = 1000, `b2` has balance = 1500
e. an error occurs

12. Suppose the following statements are executed in a client program:

```
BankAccount b3 = new BankAccount(500);
BankAccount b4 = b3;
b3.deposit(100);
```

What are the balances of `b3` and `b4` after the code is executed?

a. b3 has balance = 500, b4 has balance = 0
b. b3 has balance = 600, b4 has balance = 100
c. b3 has balance = 600, b4 has balance = 500
d. b3 has balance = 600, b4 has balance = 600
e. an error occurs

13. Suppose the following statements are executed in a client program:

```
BankAccount b3 = new BankAccount(500);
BankAccount b4 = b3;
b3 = null;
b4.deposit(b3.getBalance());
```

What are the balances of b3 and b4 after the code is executed?
a. b3 has balance = 500, b4 has balance = 500
b. b3 has balance = 0, b4 has balance = 500
c. b3 has balance = 0, b4 has balance =1000
d. b3 has balance = 0, b4 has balance = 0
e. an error occurs

14. Suppose the following statements are executed in a client program:

```
BankAccount b5 = new BankAccount(200);
BankAccount b6 = b5;
b5 = null;
b6.deposit(100);
```

What are the balances of b5 and b6 after the code is executed?
a. b6 has balance = 0
b. b6 has balance = 200
c. b6 has balance = 300
d. b6 has balance = 500
e. an error occurs

For the following exercises, consider the following Time class shown below.

```
public class Time
{
   public Time() { /* implementation not shown */ }

   public Time(int h, int m, int s) { /* implementation not shown */ }

   // resets time to myHrs = h, myMins = m, mySecs = s
   public void resetTime(int h, int m, int s) { /* implementation not shown */ }

   // advances time by one second
   public void increment(){ /* implementation not shown */ }

   // returns 1 if this time is earlier than t, returns 0 otherwise
   public int lessThan(Time t) { /* implementation not shown */ }

   public int getSecs() { /* implementation not shown */ }

   private int myHrs;
   private int myMins;
   private int mySecs;
  }
```

15. Which of the following is a **false** statement about the methods?

a. There are no static methods in this class.
b. `resetTime()` is not a constructor.
c. `Time()` is a default constructor.
d. `h` is used as the name of a formal parameter.
e. All of the statements are true.

16. Which of the following represents correct implementation for the constructor with parameters in the `Time` class above?
a.
```
myHrs = 0;
myMins = 0;
mySecs = 0;
```

b.
```
h = myHrs;
m = myMins;
s = mySecs;
```

c.
```
resetTime(h, m, s);
```

d.
```
h = 0;
m = 0;
s = 0;
```

e.
```
Time = new Time(h, m, s);
```

17. Which of the following are correct implementations for the `increment` method?

I.
```
mySecs = mySecs + 1;
```

II.
```
mySecs = getSecs() + 1;
```

III.
```
this.mySecs = this.mySecs + 1;
```

a. I only
b. II only
c. III only
d. I and III only
e. I, II, and III

Free Response Questions

1. An `Employee` object has a name (a `String`) and a salary (a `double`). An incomplete class definition is below.

```
public class Employee
{
      // code goes here for exercise a below

      public Employee(String name, double moneyEarned)
      {
            myName = name;
            mySalary = moneyEarned;
      }

      public double getSalary() {   return mySalary;  }

      public void paycut(double byPercent)
      {
            // code goes here for exercise b below
      }

      private String myName;
      private double mySalary;
}
```

a. Write a default constructor for the `Employee` class that initializes the employee's name to the empty string and his/her salary to 45000.

b. Write a method `paycut` for the `Employee` class. The method `paycut` will decrease the employee's salary by a given percentage passed as a parameter to the method.

Sample usage within a client program:

```
Employee worker1 = new Employee("Smith", 100000);
worker1.paycut(20);
// Smith now has a salary of $80000 since subtracting 20000 is 20% of 100000
```

Use the method header below to write `paycut`.

```
public void paycut(double byPercent)
{




}
```

2. An **incomplete** definition for the `Book` class is below.

```
public class Book
{
      // constructors found here

      public double getPrice() {    return price;     }

      // code goes here for exercise a below

      public void doublePrice()
      {
            // code for answer to part b goes here
      }

      private double myPrice;
}
```

a. Write a modifier method named `setPrice` for the `Book` class that allows a client program to modify the price of a book.

b. Write a method `doublePrice` that modifies the price of the book by doubling it. Use the method header below to write `doublePrice`.

Sample usage within a client program:

```
Book favoriteBook = new Book(10.00); // price of book is 10
favoriteBook.doublePrice();          // price of book is now doubled to 20
System.out.println("The new price of the book is " + favoriteBook.doublePrice());
// Output would be            The new price of the book is 20

public void doublePrice()
{









}
```