Name _____ Period_____

1. Refer to the code below,

```
public class Phone {
         private int areaCode;
         private int prefix;
         private int lineNumber;
         public Phone(int ac, int p, int ln) {
                  areaCode = ((ac > 0 && ac < 1000) ? ac : 555);
                  prefix = ((p > 0 && p < 1000) ? p : 555);
                  lineNumber = ((ln > 0 && ln < 10000) ? ln : 5555);
         }
         public String makeCall(Phone p) {
                  return "Dialing " + p.toString();
         }
         public String toString() {
                  return "" + areaCode + "-" + prefix + "-" + lineNumber;
         }
}

public class CellPhone extends Phone {
    private double longitude;
    private double latitude;
    public CellPhone(int ac, int p, int ln, double lat, double lng) {
        super(ac, p, ln);
        latitude = lat;
        longitude = lng;
    }
    public void updateLocation() {
        // uses GPS to get the updated values for longitude and latitude
    }
    public double getLongitude() {
        return longitude;
    }
    public double getLatitude() {
        return latitude;
    }
    public String toString() {
        String s = super.toString();
        return s + "; (" + longitude + ", " + latitude + ")";
    }
}

public class PayPhone extends Phone {
    private double cost;
    protected double moneyInserted;
    public PayPhone(int ac, int p, int ln, double c) {
        super(ac, p, ln);
        cost = ((c >= 0) ? c : 0);
    }
    public void insertMoney(double money) {
        moneyInserted += money;
    }
    public String makeCall(Phone p) {
        if (moneyInserted >= cost) {
            moneyInserted -= cost;
            return super.makeCall(p);
        }
        return "Please insert more money to place a call";
    }
}
```

(a) What is/are the parent class(es) associated with the PayPhone class?

Object, Phone

/1

(b) For each of the following (i) Indicate whether the statement is valid (V) or invalid (I) (ii) If the statement is not valid, indicate why.

Statement	V/I	If "I", indicate why.
`Object o = new Phone(317, 555, 1000);`	**V**	
`CellPhone cp = new Phone(459, 555, 1022);`	**I**	**A Phone is not a CellPhone**
`PayPhone pph = new CellPhone(333, 555, 4242, 23.423, 54.343);`	**I**	**A PayPhone is not a CellPhone**
`Phone ph = new CellPhone(888, 555, 6642, 78.44, 66.3);`	**V**	
`Object o = new PayPhone(954, 555, 4242, .25);`	**V**	
`PayPhone pph = new PayPhone(123, 555, 5555, 28.44, 45.6);`	I	Wrong parameters

/6

(c) What is the value of s after the code block below? Do not include quotes in your answer.

```
Phone ph = new CellPhone(444, 555, 6666, 1.2, 2.4);
String s = ph.toString();
```

444-555-6666; (2.4,1.2)

/1

(d) What is the value of s after the code block below? Do not include quotes in your answer. If an error occurs write "ERROR" AND indicate why the error occurs.

```
PayPhone pph = new PayPhone(311, 555, 6464, .25);
Phone ph = pph;
ph.insertMoney(.50);
System.out.print(pph.moneyInserted)
```

ERROR; insertMoney is not in Phone

/2

(e) Refer to the code block below to indicate what is printed for each of the following statements. If an error occurs write "ERROR" AND indicate why the error occurs.

```
Phone p = new Phone(765,999,1234);
CellPhone cp = new CellPhone(858,346,6430, 40.427437, -86.916979);
PayPhone pp = new PayPhone(212,840,9623,0.5);
```

(i) `System.out.println(p);`

765-999-1234

(ii) `Sysem.out.println(cp);`

858-346-6430; (-86.916979, 40.427437)

(iii) `System.out.println(pp);`

858-346-6430; (-86.916979, 40.427437)

(iv) `System.out.println(p.makeCall(cp));`

Dialing 858-346-6430; (-86.916979, 40.427437)

(v) `System.out.println(cp.makeCall(p));`

Dialing 765-999-1234

(vi) `System.out.println(pp.makeCall(p));`

Please insert more money to place a call

(vii)
```
pp.insertMoney(0.5);
System.out.println(pp.makeCall(p));
```

Dialing 765-999-1234

(viii) `System.out.println(cp.latitude);`

ERROR: java: latitude has private access in CellPhone

(ix) `System.out.println(cp.getLatitude());`

40.427437

(x)
```
p = cp;
System.out.println(p);
```

858-346-6430; (-86.916979, 40.427437)

/10

2. Refer to the code below,

```
class A {
    public A() {
        System.out.println("Inside A's constructor");
    }
}
class B extends A {
    public B() {
        System.out.println("Inside B's constructor");
```

```
    }
}
class C extends B {
    public C() {
        System.out.println("Inside C's constructor");
    }
}
public class Inheritance {
    public static void main(String[] args) {
        /** Statements for questions go here **/
    }
}
```

(a) After executing the statement `B b = new B();`, what is output by the program?

Inside A's constructor
Inside B's constructor

/1

(b) After executing the statement `A a = new C();`, what is ouptut by the program?

Inside A's constructor
Inside B's constructor
Inside C's constructor

/1

(c) What is the output of the following statement? System.out.println((new B()) instanceof A);

Inside A's constructor
Inside B's constructor
true

/1

(d) What is the output of the following statement? System.out.println((new C() instanceof A);

Inside A's constructor
Inside B's constructor
Inside C's constructor
true

/1

3. The `Parrot` class represents a parrot with an age in years and the ability to learn sounds which it can repeat back when asked to speak. The declaration of the `Parrot` class is shown below.

```
public class Parrot
{

 /** Constructs a new Parrot object */
 public Parrot(String name)
 { /* implementation not shown */ }

 /** @return the age of the parrot in years */
 public int getAge()
 { /* implementation not shown */ }

 /** Adds sound to the list of sounds the parrot can make
 *  @param sound the sound to add */
 public void train(String sound)
 { /* implementation not shown */ }

 /** @return a random sound that the parrot can make */
 public String speak()
 { /* implementation not shown */ }

 // There may be instance variables, constructors, and methods that are
not shown.
}
```

A pirate parrot is a type of parrot. A pirate parrot knows how to make the sound “Polly want a cracker” immediately upon birth. A pirate parrot can also steal souls whose age becomes part of the pirate parrot’s age. A pirate parrot is represented by the PirateParrot class, which you will write.

Assume that the following code segment appears in a class other than `PirateParrot`. The code segment shows an example of using the `PirateParrot` class.

```
PirateParrot polly = new PirateParrot("Polly");

System.out.println(polly.getAge()); // prints 0
/* code to increase Polly's age by 5 years */
System.out.println(polly.getAge()); // prints 5

polly.stealSoul(5);
polly.stealSoul(10);
System.out.println(polly.getAge()); // prints 20

polly.train("Walk the plank");
polly.train("Off with his head");
```

```
// Polly retires from his life as a pirate to a cushy life as a pet

Parrot myPetPolly = polly;
System.out.println(myPetPolly.getAge()); // prints 20
myPetPolly.train("Time for bed");
System.out.println(myPetPolly.speak());

/* prints one of the following, chosen at random:
 * Polly want a cracker
 * Walk the plank
 * Off with his head
 * Time for bed
 */
```

(a) Write the `PirateParrot` class. Your code must produce the indicated results when invoked by the code given above.

```
public class PirateParrot extends Parrot
{
     /** The total number of years the pirate parrot has stolen */
     private int yearsStolen = 0;
     public PirateParrot(String name)
     {
          super(name);
          train("Polly want a cracker");
     }
     public int getAge()
     {
          return super.getAge() + yearsStolen;
     }
     public void stealSoul(int soulAge)
     {
          yearsStolen += soulAge;
     }
}
```

/8