**Inheritance and Polymorphism Worksheet Name:**

**AP Computer Science**

1. Reference Semantics Mystery

The following program produces 4 lines of output. Write the output below, as it would appear on the console.

public class BasicPoint {

int x;

int y;

public BasicPoint() {

x = 2;

y = 2;

}

}

public class ReferenceMystery {

public static void main(String[] args) {

int a = 7;

int b = 9;

BasicPoint p1 = new BasicPoint();

BasicPoint p2 = new BasicPoint();

addToXTwice(a, p1);

System.out.println(a + " " + b + " " + p1.x + " " + p2.x);

addToXTwice(b, p2);

System.out.println(a + " " + b + " " + p1.x + " " + p2.x);

}

public static void addToXTwice(int a, BasicPoint p1) {

a = a + a;

p1.x = a;

System.out.println(a + " " + p1.x);

}

}

2. Inheritance Mystery

Assume that the following classes have been defined:

|  |  |
| --- | --- |
| public class A extends B {  public void method2() {  System.out.print("a 2 ");  method1();  }  }  public class B extends C {  public String toString() {  return "b";  }    public void method2() {  System.out.print("b 2 ");  super.method2();  }  } | public class C {  public String toString() {  return "c";  }    public void method1() {  System.out.print("c 1 ");  }    public void method2() {  System.out.print("c 2 ");  }  }  public class D extends B {  public void method1() {  System.out.print("d 1 ");  method2();  }  } |

Given the classes above, what output is produced by the following code?

C[] elements = {new A(), new B(), new C(), new D()};

for (int i = 0; i < elements.length; i++) {

System.out.println(**elements[i]**);

elements[i].**method1**();

System.out.println();

elements[i].**method2**();

System.out.println();

System.out.println();

}

**3. Polymorphism Mystery**  
Assume that the following classes have been defined:

|  |  |
| --- | --- |
| public class Pen extends Sock {  public void method1() {  System.out.print("pen 1 ");  }  }  public class Lamp {  public void method1() {  System.out.print("lamp 1 ");  }  public void method2() {  System.out.print("lamp 2 ");  }  public String toString() {  return "lamp";  }  } | public class Book extends Sock {  public void method2() {  System.out.print("book 2 ");  super.method2();  }  }  public class Sock extends Lamp {  public void method1() {  System.out.print("sock 1 ");  method2();  }  public String toString() {  return "sock";  }  } |

Given the classes above, what output is produced by the following code?

Lamp[] elements = {new Book(), new Pen(), new Lamp(), new Sock()};

for (int i = 0; i < elements.length; i++) {

System.out.println(elements[i]);

elements[i].method1();

System.out.println();

elements[i].method2();

System.out.println();

System.out.println();

}

4. Inheritance Mystery

Assume that the following classes have been defined:

|  |  |
| --- | --- |
| public class Brian extends Lois {  public void b() {  a();  System.out.print("Brian b ");  }  public String toString() {  return "Brian";  }  }  public class Lois extends Meg {  public void a() {  System.out.print("Lois a ");  super.a();  }    public void b() {  System.out.print("Lois b ");  }  } | public class Meg {  public void a() {  System.out.print("Meg a ");  }  public void b() {  System.out.print("Meg b ");  }  public String toString() {  return "Meg";  }  }  public class Stewie extends Brian {  public void a() {  super.a();  System.out.print("Stewie a ");  }  public String toString() {  return super.toString()+" Stewie";  }  } |

Given the classes above, what output is produced by the following code?

Meg[] elements = {new Lois(), new Stewie(), new Meg(), new Brian()};

for (int i = 0; i < elements.length; i++) {

**elements[i].a();**

System.out.println();

**elements[i].b();**

System.out.println();

System.out.println(**elements[i]**);

System.out.println();

}