1 Confusing Constructors

What is the output of the following program after we execute the main method?

```java
public class Confusing {
    private Confusing(Object o) {
        System.out.println("Data Structures");
    }
    private Confusing(double[] dArray) {
        System.out.println("Algorithms");
    }
    public static void main(String[] args) {
        int[] array = new int[4];
        IntList list = IntList.list(array);
        Confusing Antares = new Confusing(array);
        Confusing Christine = new Confusing(list);
        Confusing Nicolas = new Confusing(null);
    }
}
```

Data Structures
Data Structures
Algorithms (java uses the most specific constructor)
2 The ABCs of OOP

Indicate what each line the main program in class D would print, if the line prints anything. If any lines error out, identify the errors as compile-time or runtime errors and cross out the corresponding lines.

```java
public class A {
    public void x() { System.out.println("Ax"); }
    public void y(A z) { System.out.println("Ay"); }
}

public class B extends A {
    public void y() { System.out.println("By"); }
    public void y(B z) { System.out.println("Byz"); }
}

public class C extends A {
    public void x() { System.out.println("Cx"); }
}

public class D {
    public static void main(String[] args) {
        A e = new B();
        A f = new C();
        B g = new A(); // Compile-Time Error. A is not a subclass of B
        B h = new C(); // Compile-Time Error. Although B and C are both
                        // children classes of A, B and C are not related to each other.
        C i = (C) new A(); // Runtime Error. Casting would trick the com-
                            //piler to think of the new object as type C and then assign it to i. While
                            //running the program, casting will crash because the new object is type A
                            //in dynamic binding, which cannot be assigned to class C (as A is not a sub-
                            //class of C).
        B j = (A) new C(); // Compile-Time Error. Casting will trick the
                            //compiler to think of the new object as type A. However in run-time when look-
                            //ing at the dynamic types, we cannot assign it to type B since A is not a
                            //subclass of B.
        B k = (B) e; // e is type B in dynamic type, so the assignment works
                     //out fine in run-time.

        f.x(); // Cx
    }
}
```
3 Fix this Waffle Code

Given the following interface and classes, fill in the blanks below so that the code compiles.

```java
public interface Edible {
    void eat();
}

public abstract class Food implements Edible {
    public abstract void cook();
}

public class Pancake extends Food {
    public void eat() {
        System.out.println("Pancake");
    }
    public void cook() {
        System.out.println("Made Pancake!");
    }
}

public class Waffle implements Edible {
    public void eat() {
        // Fill in the blank here
    }
}
```
System.out.println("Waffle");

public static void main(String[] args) {
    Edible arr = new Edible[2];
    arr[0] = new Pancake();
    arr[1] = new Waffle();
}