Name _____ Section Number _____

CS110 Exam #4 Practice
Sections 1-3 *** TURN OFF ALL ELECTRONIC DEVICES*** Bob Wilson

You may use your crib sheet (one 8-1/2 x 11 page both sides) and only your crib sheet.

1. Expression Evaluation (20 Points)

```
public static void main(String [] args)
{
    int a = 5, b = 13;
    double c = a - b;
    String [] s = {"0", "5", "9", "13"};
    char d = 'd';                                   // What prints?

    System.out.println(s[0] + c);                   // a. _________

    System.out.println(a + b + s[1]);               // b. _________

    System.out.println(s[2] + a + b);               // c. _________

    System.out.println(s[3].length());              // d. _________

    System.out.println(s.length);                   // e. _________

    System.out.println((char) (d + 3));             // f. _________

    System.out.println(d - 'a');                    // g. _________

    System.out.println(factorial(b % a));           // h. _________

    System.out.println(a++ + b++);                  // i. _________

    System.out.println(++a + ++b);                  // j. _________
}

private static int factorial(int n)
{
    int result = 1;
    while (n > 1)
        result *= n--;
    return result;
}
```

2. Selection Statements (20 points)
What is printed by the following code segment?

```
String [] strings = {"Okay", "Nope", "Maybe", "Null"};
boolean alpha = true;
boolean beta = false;
char zero = '0';

                                                    // Printed? (Yes or No)
if (strings[1].charAt(1) != strings[3].charAt(1))
      System.out.println("It's all Greek to me"); // a._________________

if (strings[0].length() == strings[2].length()) {

   if (alpha && beta)
       System.out.println("Gamma");               // b._________________

   else
       System.out.println("Delta");               // c._________________
}

else if (beta)
      System.out.println("Epsilon");              // d._________________

else if (alpha)
      System.out.println("Phi");                  // e._________________

else
      System.out.println("Pi");                   // f._________________

switch (zero)
{
      case '0':
            System.out.println("Eta");            // g._________________
            break;

      case 0:
            System.out.println("Rho");            // h._________________
            break;

      default:
            System.out.println("Tau");            // i._________________
            break;
}

if (zero != 0)
      System.out.println("Omega");                // j._________________
```

3. Loops

What pattern is printed by the following code?

```
public class Test3
{
  public static void main(String[] args)
  {
    for (int i = 0; i < 6; i++)
    {
      int j = 0;
      while (j < 6)
      {
        if (j < i)
          System.out.print("X");
        else if (i < j)
          System.out.print("Y");
        else
          System.out.print("Z");
        j++;
      }
      System.out.println();
    }
  }
}

> java Test3
```

4. Arrays of Objects and Exceptions (20 points)

Study the following code and explain below what is printed:

```
String [] strings = new String[5];

try {
   strings[1] = "Hello";
   strings[2] = "Goodbye";
   strings[3] = "Konichiwa";
   strings[4] = "Sayonara";
}
catch (ArrayIndexOutOfBoundsException e) {
   System.out.println("Oops" + e);
}

try {
   for(String s : strings)
      System.out.println(s);
}
catch (NullPointerException e) {
   System.out.println("Ouch" + e);
}
finally {
   System.out.println("Done");
}

for (int i = 1; i < strings.length; i++)
   System.out.println(strings[i].charAt(i));

for (String s : strings)
   System.out.println(s.length());
```

Show here what is printed by the above lines of code and explain what happened:

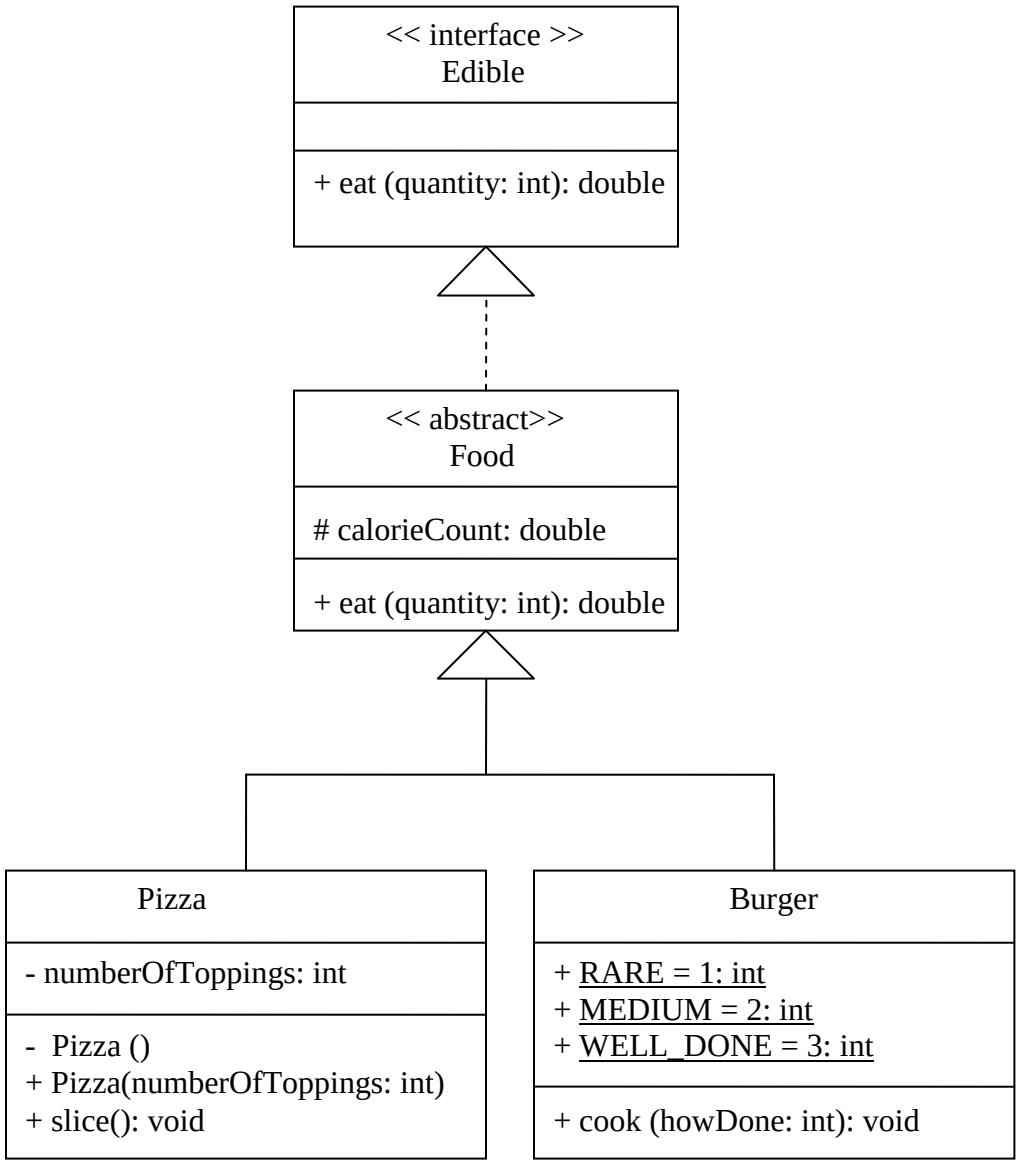
5. Classes, Interfaces, Inheritance, and Polymorphism (20 Points)

Using the attached UML Diagram, answer if each of these lines of Java code in class FastFood is valid or not. **If not, explain why not.**
(Note: Assume that the FastFood class is **not** in the same package as the other classes.)

```
public class FastFood
{
  public static void main(String[] args)
  {
    Food myFood;
    Pizza myPizza;
    Burger myBigMac;

    myFood = new Pizza();                    // a. Valid? _____


    myFood = new Pizza(3);                   // b. Valid? _____


    myPizza = myFood;                        // c. Valid? _____


    myFood = new Burger();
    myBigMac = (Burger) myFood;              // d. Valid? _____


    myFood.cook(Burger.RARE);                // e. Valid? _____


    System.out.println(myFood.eat(2));       // f. Valid? _____


    ((Burger)myFood).cook(Burger.RARE);      // g. Valid? _____


    ((Pizza)myFood).slice();                 // h. Valid? _____


    System.out.println(myFood.calorieCount); // i. Valid? _____


    Edible goodToEat = myFood;               // j. Valid? _____

  }
}
```

UML Diagram for Question 5:

Note: You can tear this sheet off and use it for reference while answering question 5.



Solutions:

1. Expression Evaluation

a. 0-8.0
b. 185
c. 9513
d. 2
e. 4
f. g
g. 3
h. 6
i. 18
j. 22

2. Selection Statements

a. Yes
b. No
c. No
d. No
e. Yes
f. No
g. Yes
h. No
i. No
j. Yes

3. Loops

```
> java Test4
ZYYYYY
XZYYYY
XXZYYY
XXXZYY
XXXXZY
XXXXXZ
```

4. Arrays of Objects and Exceptions

```
> java Test4     (no ArrayIndexOutOfBoundsException occurs)
null             (strings[0] is not accessed with a ".")
Hello
Goodbye
Konichiwa
Sayonara
Done             (no exception occurs just the finally clause)
e                (strings[0] is not accessed with .charAt()
o
i
n
java.lang.NullPointerException  (caused by strings[0].length())
```

5. Classes, Interfaces, Inheritance, and Polymorphism

```
public class FastFood
{
  public static void main(String [] args)
  {
    Food myFood;
    Pizza myPizza;
    Burger myBigMac;

    // myFood = new Pizza();              // a. Valid? __NO_______
    // Error: Default constructor in Pizza is private

    myFood = new Pizza(3);                // b. Valid? __YES______


    // myPizza = myFood;                  // c. Valid? __N0_______
    // Error: incompatible types (narrowing conversion needs a cast)

    myFood = new Burger();
    myBigMac = (Burger) myFood;           // d. Valid? __YES______


    // myFood.cook(Burger.RARE);          // e. Valid? __NO_______
    // Error: cannot find symbol: method cook(int), location: Food

    System.out.println(myFood.eat(2));    // f. Valid? __YES______


    ((Burger)myFood).cook(Burger.RARE);   // g. Valid? __YES______


    // ((Pizza)myFood).slice();           // h. Valid? __NO_______
    // ClassCastException: Burger at FastFood.main(FastFood.java:31)

    // System.out.println(myFood.calorieCount); // i. Valid? __NO_
    // Error: calorieCount is protected in Food

    Edible goodToEat = myFood;            // j. valid? __YES______

  }
}
```