ROUND ONE

Question 1. Write the body of the following method.

// Return the longest string in array A
// Assume that A contains at least one String
public static String longest(String[] A) {
    \textit{MISSING CODE}
}

Question 2. Write a method named \texttt{getPosInt} that works as follows:

- Ask the user to type in a positive int value.
- If the value typed in is not positive, print an error message and ask again for a positive value.
- Repeat this until a positive value has been typed in. Then return that value.

You do not need to handle the case where the user types in a value that is not an integer.

Question 3. Consider the following code:

\begin{verbatim}
Person p1 = new Person("Sandy", 12);
Person p2 = new Person("Amelia");
if (p1.getAge() != p2.getAge()+12) {
    System.out.println("error!");
}
\end{verbatim}

Write a complete definition of a \texttt{Person} class so that this code would compile and execute, and would not print “error”.

Question 4. Complete the \texttt{specialMenu} method of the \texttt{Restaurant} class below.

\begin{verbatim}
public class Restaurant {
    private ArrayList<String> appetizers;
    private ArrayList<String> mainCourses;
    private ArrayList<String> desserts;
    // return an ArrayList of size 3 containing the names of an
    // appetizer, a main course, and a dessert, each chosen
    // randomly from the appropriate field of this class;
    // assume that the three fields each contain at least 1 string
    public ArrayList<String> specialMenu() {
        \textit{MISSING CODE}
    }
}
\end{verbatim}
**Question 5.** Assume that a Student class has been defined with the following fields and methods:

```java
public static final int UGRAD = <some value>;
public static final int GRAD = <some value>;
public static final int SPECIAL = <some value>;
public static final int FULLTIME = <some value>;
public static final int PARTTIME = <some value>;

private int kind; // has value UGRAD, GRAD, or SPECIAL
private int status; // has value FULLTIME or PARTTIME
private int credits;

public int getKind() { return kind; }
public int getCredits() { return credits; }
public void setStatus(int st) { status = st; }
```

A student’s status is supposed to be FULLTIME if their kind is GRAD and they have at least 6 credits, or their kind is UGRAD and have at least 12 credits (SPECIAL students are always PARTTIME).

Assume you have a Student object s whose kind and credits fields have been initialized but not its status field. Write a code fragment (not a whole method) that is not part of the Student class and that sets s’s status appropriately.

**Question 6.** Assume that a Point class has been defined. Each Point stores two int values, and the Point constructor takes two int values as arguments.

Write a code fragment (not a whole method) that, when executed, creates this memory diagram:
Question 7. Complete the following method.

```java
public static String oddChars( String s ) {
    // MISSING CODE
}
```

Question 8. Assume that a Person class has been defined, including an equals method. Write the body of the following method.

```java
public static boolean hasDuplicate(ArrayList<Person> p) {
    // MISSING CODE
}
```
ROUND THREE

**Question 9.** Assume that a Person class has been defined, with a getEyeColor method that returns a lower-case String.

Complete the following method.

```java
public static boolean[] blueEyeRows(Person[][] p) {
    // MISSING CODE
}
```

**Question 10.** Consider the following code:

```java
Car aCar = new Car("Honk");
System.out.println(aCar.getHorn());
aCar.changeHorn(2);
System.out.println(aCar.getHorn());
aCar.changeHorn("Beep");
System.out.println(aCar.getHorn());
aCar.changeHorn(4);
System.out.println(aCar.getHorn());
```

Write a complete definition of a Car class so that this code would compile and execute, and would print

```
Honk
HonkHonk
Beep
BeepBeepBeepBeep
```
**Question 11.** Assume that a `Point` class has been defined. Write field, constructor, and method declarations for a `Polygon` class (just declarations no code for the bodies), according to the following specification.

- A polygon should keep track of its location, color, and number of sides.
- The location should be a `Point`.
- Colors should be int values with the names `RED` or `BLUE`. Those values should be available to be used outside the class.
- It should be possible to create a polygon given its location and number of sides, or given its location, number of sides, and color.
- A polygon should be able to draw itself (at its location).
- The polygon’s `draw` method should use a helper method that draws one side, given the color to use, and “from” and “to” points.
- A polygon should be able to determine if it is equal to another polygon.
Question 12. Assume a Person class has been defined. Fill in all missing parts of the following code to implement the Line class, which represents people standing in line.

You must use an array to represent the line of people.

```java
public class Line {
    // fields
    MISSING CODE

    // create an empty line that can hold n people
    public Line( int n ) {MISSING CODE}

    // create a line that can hold n people, and that
    // already includes all of the people in array A
    // (with the person in A[0] at the front of the
    // line, the person in A[1] next, etc). If A has
    // more than n people, just include the first n
    public Line( int n, Person[] A ) {MISSING CODE}

    // add p to the end of the line if there’s room
    // return true if p was added, else return false
    public boolean addToEnd( Person p ) {MISSING CODE}

    // remove and return the person at the front of
    // the line and move everyone else up;
    // if the line was empty, return null
    public Person removeFromFront() {MISSING CODE}

    // return the number of people in line
    public int numInLine() {MISSING CODE}
}
```