```
int i;
i = 3;
String name = new String("Fred");
char letter = name.charAt(1);
i = name.length();
```

The code above declares a number of identifiers. List all of the identifiers and indicate which ones are for primitive types and which ones refer to object types.

Describe the at least one important difference between primitive types and object types.

Complete the method below:

```
/**
  * Returns a substring that contains all of the characters up to
  * but not including the first occurence of letter.
  * For example, front("Glow worms glow", 'm'); would return
  * "Glow wor".
  *
  * @param phrase String from which a substring will be obtained
  * @param letter Character value used to determine the length of the substring
  * return a substring containing all of the characters in phrase up to
  *        but not including the first occurence of letter
  */
public static String front(String phrase, char letter) {









}
```

Complete the main method below which repeatedly asks a user to enter the integer value **71** and exists once the user enters the correct integer value. You may assume that the user always enters an integer (it just might not be **71**).

```
import javax.swing.*;

public static void main(String[] args) {



}
```

Describe the difference between a class field and an object field. Give an example of each.

Add the correct code to the following method so that the result returned by the method is as follows:

- **8** when x is greater than or equal to 3.2
- **0** when x is less than 3.2 and x and y are equal
- **3** when x is less than 3.2 and x and y are the negative of each other (e.g., x=3, y=-3)
- **-1** when for any other situation

```
public static int quiz7(double x, double y)
{
  int result;

  return result;
}
```

Recall the LaserPrinter class from lecture yesterday. Parts of the class are shown below.

```
public class LaserPrinter {

  private static final int PAGES_PER_CARTRIDGE = 10000;
  private static final int PAPER_TRAY_CAPACITY = 500;
  private double tonerLevel;
  private int paperCount;
  private int pagesPrinted;
  private int tonerReplacementCount;

  public int getTonerLevel() { // ... }

  public int getPagesInTray() { // ... }

  /**
   * Returns total number of pages printed in printer's lifetime
   * @return total number of pages printed in printer's lifetime
   */
  public int getPagesPrinted() {









  }

  public int getNumberOfTonerCartridgesConsumed() { // ... }

  public boolean printPages(int count) { // ... }

  /**
   * Logically removes the current toner cartridge and replaces
        *  it with a new toner cartridge
   */
  public void replaceToner() {









  }

  public int loadPaperTray(int count) { // ... }
}
```

Implement the getPagesPrinted and replaceToner methods.

Complete the method below:

```
/**
  * Breaks a string into four separate strings and returns an
  * array containing the four strings.  For example, if the
  * method is passed "aabbccdd" it will return an array of strings
  * where the first element in the array is "aa" the second element
  * is "bb", etc.
  * @param characters A string that is at least four characters
  *                   long and divisible by four.
  * @return an array of four strings containing the first, second,
  *         third, and fourth quarter of the string passed in.
  */
public static String[] breakIntoQuarters(String characters) {
```