Unit 4 – Worksheet 07 Creative Objects Name: _____

Directions: Time to be creative:

1. Come up with 3 ideas that you could use to write your own JAVA class to implement obects. Be sure that your idea is able to use the following concepts withing the code:

    a. Encapsulation
    b. Constructor Method
    c. Accessor Method(s)
    d. Mutator Method(s)
    e. Parameters Used
    f. Main Method
    g. Instance Variables
    h. State Variables
    i. Implements at least one default state variable
    j. Implements at least one static variable
    k. Object Reference(s)

    List three ideas here: 1) _____

    2) _____

    3) _____

2. Choose one of the ideas above to be the project that you will be working on for a while:

    My Choice (className): _____

3. Type up a plan for the class that you will be writing. For this assignment, write the class outline and structure (not the entire class).

4. The class outline and structure for the ***bankAccount*** class (Unit 4 WS06 The Bank Account) is shown on the next page. For today's assignment, you need to create a similar document for a class that you design on your own. Your work could be done in netbeans; however, can be done in a word processor.

```
public class bankAccount  {

        /** 3 Instance variables
           *   1 double variable balance, 2 String variables pw and username.      */

        public bankAccount(String a)  {

                /** Constructor Method
                   * Default state variable pw set to XYZ, Default state variable balance set to zero
                   * State variable username set to user-defined name that is sent as a parameter              */
        }

        public void editPW( )  {

                /** Mutator Method
                   * Ask and grab user input for a new password.  When done, ask the user to re-enter the new password.
                   * Pass both user entries to the verifyPW( ) method to make sure that they match.
                   * If they match, change the account password and tell the user that the password was changed.
                   * If they do not match, do not change the password.  Tell the user there was no match and no change  */
        }

         public boolean verifyPW(String a, String b){

                /** Accessor Method
                   * Use the user's new password and re-entry as parameters to check to see if they match.
                   * If they match return true.  If they do not match, return false.                                  */
        }

        public void deposit( )  {

                /** Mutator Method
                   * Ask the user for the deposit amount and add the deposit amount to the account balance          */
        }

        public void withdraw( )  {

                /** Mutator Method
                   * Ask the user for the withdraw amount and subtract the amount from the account balance         */
        }

        public double getBalance( )  {

                /** Accessor Method - return the current account balance          */
        }

        public String getPW( )  {

                /** Accessor Method – return the current account password        */
        }

        public String getUsername( ){

                /** Accessor Method – return the current username                 */
        }

        public void changeUsername( )  {

                /** Mutator Method
                   * Ask the user for a new username and change the account username               */
        }
}
```