1. Consider the following constructor:

```
public team( )  {
    wins = 0;
    losses = 0;
    games = 0;
}
```

1a. To work correctly, the constructor method must be located inside a class with what name?

1b. In order for this constructor method to work appropriately, the instance variables need to be created and are typically listed outside of the ***team( )*** method, but inside the ***team*** class. Write the code that would be needed to create the instance variables (for this problem it can be done in 1-3 lines of code).

1b. Write one line of code that could be used in a main method to create a team object (instance of team).

1c. In a particular class, a team object named Vikings was created. Write a JAVA statement that uses this team's object reference variable to access the number of wins for this instance of team.

1c. Write another method called ***wonGame( )*** that has no parameters and no return statement, but does increase both the number of wins and the number of games for any given team.

1d. Write another method called ***lostGame( )*** that has no parameters and no return statement, but does increase both the number of losses and the number of games for any given team.

1e. Write another method called ***totalGames( )*** that has no parameters and simply returns the number games that any given team has played.

1f. Write another method called ***winPercentage( )*** that has no parameters. This method should calculate calculate a team's winning percentage (a number between 0 and 100, returned as a decimal, calculated by dividing a team's number of wins by their total number of games).

1g. Now that you have written some code for this problem above, try to put it all together on a computer. Write the above methods and create a main method. Then test your work by coding various commands in the main method. Be sure to understand how to use each method and to verify that they are working correctly.

2. Below you will find the basic structure for a class called ***piggyBank*** that builds and manipulates piggyBank objects. Below it is a class called ***piggyMain*** that contains the main method (list of directions to follow). First try to write what is needed to complete the missing lines of code in these classes so that the displayed output occurs. When done, try entering your code into netbeans to test it.

```
public class piggyBank {
   double money;
   public piggyBank( )  {
                                          //code needed
   }
   public double getTotal( )  {
                                          //code needed
   }
   public void addMoney(int a)  {
                                          //code needed
   }
   public void emptyBank( ) {
                                                                  //code needed
   }
   public static void main(String[] args) {
                                                                  //code needed to create the piggyBank
      System.out.println("Bank Cretead, balance = "+bigSavings.money);
      System.out.println("Welcome to your piggy bank!");
      Scanner getChoice = new Scanner(System.in);
      int choice=0;
      while(choice!=7)
      {
          System.out.println("Current bank balance = "+bigSavings.getTotal());
           System.out.println("What would you like to do?");
           System.out.println("1. Show current money");
           System.out.println("2. Add a penny");
           System.out.println("3. Add a nickle");
           System.out.println("4. Add a dime");
           System.out.println("5. Add a quarter");
           System.out.println("6. Empty bank");
           System.out.println("7. Quit Program");
            System.out.print("Enter Choice:  ");;
            choice = getChoice.nextInt();
            if(choice==1)
                                                                  //code needed
            else if(choice==2 || choice==3 || choice==4 || choice==5)
                                                                  //code needed
            else if(choice==6)
                                                                  //code needed
            else if(choice==7)
                                                                  //code needed
             else
                System.out.println("You did not enter a valid option!");
       }
    }
}
```

Output (If the user chooses option 6)

Your bank has been emptied!
Current bank balance = 0.0
What would you like to do?
1. Show current money
2. Add a penny
3. Add a nickle
4. Add a dime
5. Add a quarter
6. Empty bank
7. Quit Program
Enter Choice: