EECS 1510: Intro to Object Oriented Programming
Project 4: Control Structures
140 Points Due in class, Thursday March 13, 2014

Submission Procedure.   For the submission of this project, be sure to read the posted file "Printed Submission of Projects.doc".

For the printout of each program, give *two* sample runs, one using the values from a sample dialog below, the other using values from some "special" case.

A special case might be entering a negative integer for a grade, having three oldest persons, or a null string for a line of text. Your program must handle the special case you use.

Each program must also follow the 8 program standards given below.

Program 1. Using JOptionPane (40 points).

Finding the oldest person in a large family can be a tedious task. Unfortunately, that's exactly what your dad has asked you to do. He will give you a list of your relatives with their ages, and he expects you to give him the name of the oldest person.

Write a program, using JOptionPane, that reads in a list of names and ages until the user enters "quit" as a person's name. Use JOptionPane to prompt for the name and then prompt again for the age, until the name "quit" is given. When the sentinel (quit) appears as the name, your program should display the name of the oldest person and then terminate. For example, with the list of entries

```
Tommy
17
Don Hodges
42
Cathy

Ken Summers
19
Elsa Turner
35
quit
```

the program will print

```
The oldest person is Kathy.
```

Name this program **Oldest.java.** For the normal printout for this program, give one screenshot of a name as input, one screen with the age as input, and one screen that shows the oldest as output.

Note: The method `showInputDialog` in `JOptionPane` returns a string for the value entered by the user. To convert the string to an integer, use the wrapper class Integer. For example: with

```
int I;    String inputString;
```

You can use

```
I  = Integer.parseInt(inputString);
```

Program 2. The Little Parking Garage (30 points).

A parking garage has 5 customers daily. The garage charges a $5.00 minimum fee to park up to two hours. The garage charges an additional $1.00 per hour for each hour (or part of an hour) over two

hours. The maximum charge for any given day is $12.00. All cars are gone by midnight. Write a program to calculate and print a summary of the charges for a day. For input the program will read the hours of usage for each of the 5 cars. The program will print the result, as shown below.

```
Enter the hours parked for car 1:   1.5
Enter the hours parked for car 2:   4.1
Enter the hours parked for car 3:   12.0
Enter the hours parked for car 4:   9.3
Enter the hours parked for car 5:   0.5

Total Hours 27.4
Total Charge $42.00
```

Use one decimal place for the hours parked, and 2 for the total charge. Name this program **ParkingGarage.java**

Program 3. Counting Change (30 points) Write a Java program that performs the following task. A user is to be prompted for an integer value sum that is supposed to be in the interval 0 . . 99. If the value of sum is not in the proper interval, the only actions that should take place are displaying an error message and returning control back to the operating system.

If the value for sum is in the proper interval, the user should be prompted for a number of dimes d, a number of nickels n, and a number of pennies p. If the monetary value of the coins provided by the user equals sum, display the message "Yes" and exit the program. Otherwise display the message "No" and prompt the user one more time to enter a correct amount of coins. Name this program **CountingChange.java**

Extra Credit (10 points): Use a static method to prompt the user for the coins and return the monetary value of the coins.

Program 4. Secret Number (40 points) Write a program to generate a random number between 1 – 100 and keep it as a secret number. The program will then check if a user can guess the secret number. The user can continue guessing the number until the number is found or the user can enter 0, which will terminate the program.

Each time the user makes a guess, the program will report as below:
     1. Way Too High or Way Too Low (more than 30 off)
     2. High or Low (between 10 and 30 points off)
     3. A Little High or A Little Low (less than 10 points off)
If secret number is 74 and the user enters 26, the program will print "Way too Low". If the user then says 65, then the program will print "A Little Low". Name this program **SecretNumber.java**

NOTE: The 8 program standards are given below:

1. Program Header . In addition to you name and project number, include a brief description of the program functionality in the header comment for the program

2. Variables and Constants. All variables and constants in a program must be declared at the beginning of the program

3. **Initialization.** Variables may not be initialized in their declaration.

4. **Printing of `if-then`.** Use one of the following
```
if (expression)   statement

if (expression)
   Single-statement

if (expression) {
   Multiple-statements
}
```

5. **Printing of `if-then-else`.** Use one of the following
```
if (expression) statement
else            statement

if (expression)
   Single-statement
else
   Single-statement

if (expression) {
   Multiple-statements
} else {
   Multiple-statements
}
```

6. **Printing of `while` loops.** Use one of the following
```
while (expression)   statement

while (expression)
   Single-statement

while (expression)  {
   Multiple-statements
}
```

7. **For loops.** The bodies of all for-loops must be indented at least 3 (but no more than 5) spaces:
```
for(int i = 0; i < n; i++) {
   System.out.println("Enter the number: );
   num = stdin.nextInt();
   if (num < 0)      num_neg++;
   else if (num > 0) num_pos++;
   else              num_zero++;
}
```

8. **Methods.** The bodies of all functions must be indented at least 3 (but no more than 5) spaces:
```
public static double CircleArea(double r) {
   const double Pi = 3.1415;
   return Pi * r * r;
}
```