Department of Computer Science
Purdue University, West Lafayette

Fall 2011: CS 180 Problem Solving and OO Programming
Exam 1 Solutions **Wednesday October 19, 2011.**

Q 1 Answer the questions below assuming that binary integers are represented in 5-bits using 2's complement. The leftmost bit indicates the sign of the number. For real numbers simply provide your answer using as many bits needed for the integer part of the number and only 4-bits for the fraction part of the number. You do not need to normalize the number or represent it in the IEEE format.

(a) Binary equivalent of 12_{10}= 01100

(b) Decimal equivalent of 11111_2=-1

(c) Binary sum of $0011_2 + 0100_2$=0111

(d) Binary equivalent of 9.7_{10}=1001.1011

(e) Binary equivalent of hexadecimal number $3A_{16}$= Cannot be represented in 5 bits; otherwise 0111010

(f) Largest integer (decimal value)=15

(g) Smallest integer (decimal value)=-16

Q 2 You are required to write a Java program that performs the following tasks in the sequence specified below.

(a) Prompt the user to enter a number and read the number typed. The user may type any valid number.

(b) Prompt the user to enter another number and read the number typed. Again, the user may type any valid number. [5 points]

(c) Prompt the user for the operation to be performed. The user will select one of two operations: add (+) or multiply (*). The user will type a + if an add operation is to be performed and a * if a multiply operation is to be performed.

(d) Perform the operation specified by the user on the two numbers the program has already input.

(e) Display the result of applying the operation. The result must be a number of type `double` and displayed to only 2-digit accuracy and displayed as indicated in the examples below. Note that all numbers are displayed to 2-digit accuracy.

Here is an example of how your program should interact with the user. 20.0, 3.5, and * are sample inputs provided by the user in the first example below. Use the prompts given below. All else on the lines below is output generated by your program. Ignore the possibility that the user input may make your program result in an overflow or underflow.

```
Enter a number: 20.0
Enter another number:3.5
Enter operation to be performed (+ or *): *
20.00*3.50=70.00
```

And below is another example.

```
Enter a number: -3.3356
Enter another number: 1
Enter operation to be performed (+ or *): +
-3.34+1.00=-2.34
```

Listing 1: Program for Q2. (`Calculate.java`)

```
1
2
3  import java.util.*;
4  public class Calculate{
5
6      public static void main(String [] args){
7          double num1, num2; // To hold input numbers.
8          String opString;// To hold operator.
9          char op; // To hold character corresponding to the operator.
10         Scanner s=new Scanner(System.in);
11         System.out.print("Please enter a number: ");
12         num1=s.nextDouble();// Get the first number,
13         System.out.print("Please enter another number: ");
14         num2=s.nextDouble(); // and the second one.
15         System.out.print("Please enter an operator (+ or *): ");
16         opString=s.next(); // There is no nextChar() method.
17         op=opString.charAt(0); // Extract the operator from the string.
18
19         double result=0; // Initialize result to avoid compile error.
20         if(op=='+'){
21             result=num1+num2; //Add
22         }
23         if(op=='*'){
24             result=num1*num2; // Multiply
25         }
26         // Output formatted result. If the operator is
27         // not * or + then a 0 will be output.
28         System.out.printf("%.2f%c%.2f = %.2f\n", num1, op,
29                           num2,  result);
30     }// End of main()
31 }// End of Calculate
```

Q 3 You are required to write a Java program that reads lines of text from the console one after the other. For each line read, the program checks if the first and the last character on that line are the same. If they are *not* the same the program ignores that line, If the first and the last characters on the line read *are* the same then the program prints *the first* character on that line on the console and terminates. The example below should help you better understand what your program is supposed to do.

Input read from the console	**Program action**
How do you do?	Do not print this line because the first and the last characters on this, i.e., 'H' and 'o', line are not the same.
CS 180 is challenging!	Do not print this line because the first and the last characters, i.e., 'C' and '!', are not the same.
*ABC%	This line is also not to be printed as the first and the last characters are not the same.
Sator Arepo Tenet Opera RotaS	The first and the last characters are the same. Hence print the character S and terminate.

Note again that your program must continue to read lines of text from the input until the user types a text in which the first and the last characters are identical.

Listing 2: Program for Q3. (`FirstLastChar.java`)

```
1
2
3  import java.util.*;
4  public class FirstLastChar{
5
6      public static void main(String [] args){
7          String line=""; // Holds line input.
8          Scanner s=new Scanner(System.in); // Create a Scanner object.
9          char first=' '; // Holds first character.
10         char last=' '; // Holds second character.
11         boolean done=false; // Indicates if reading should stop.
12         while(!done){
13             line=s.nextLine(); // Read next line
14             if(line.length()!=0){ // No action if the line empty.
15                 first=line.charAt(0); // Extract first char and
16                 last=line.charAt(line.length()-1);// the last char
17                 if(first==last&& first!=' '){ // Ignore space at the start
18                     done=true;
19                 }
20             }// End of empty line check.
21         }// End of input reading loop.
22         System.out.println(first); // Display the first character
23     }// End of main()
24 }// End of FirstLastChar
```

<End of Exam 1 Solutions>