97/90 = (108/100)

CS 1401, Exam #2, TR version

Date: Tuesday, October 14, 2014
Name (please type legibly, ideally in block letters): [redacted]

On October 14, 1066, the invading Norman army defeated the English king at the Battle of Hastings. Eventually, the conquerors mixed with the local population and formed the modern English nation.

1-2. Every year on October 14, British historians have seminars devoted to the Battle of Hastings. Use a loop to write a piece of code that asks the user for the starting and the ending years and then prints statements about the seminars which occurred during these years. For example, if the user inputs 1967 and 1970, your piece of code should print the following sequence of statements:

20+7/20

```
On October 13, 1967, there was a seminar devoted to the Battle of Hastings.
On October 13, 1968, there was a seminar devoted to the Battle of Hastings.
On October 13, 1969, there was a seminar devoted to the Battle of Hastings.
On October 13, 1970, there was a seminar devoted to the Battle of Hastings.
```

For extra credit: improve the previous program by taking into account that there were no seminars during the second world war, i.e., between 1939 and 1944. As a result, if the starting year is 1938 and the ending year is 1946, you should only print statements corresponding to 1938, 1945, and 1946.

```
// Previous code information, including Scanner information
    Scanner input = new Scanner (System.in);
    System.out.println ("Please enter first year: ");
        int starting_year = input.nextInt(); // first year is generated
    System.out.println ("Now, enter final year.");
        int final_year = input.nextInt(); // second year is generated
for ( i = starting_year; i <= final_year; i++) // loop from starting year to ending year is created
    {
    System.out.println ("On October 13, " + i + ", there was a seminar devoted to the Battle of Hastings);
                                                // list is printed
    } // end of loop
// later code information
```

Extra credit:

```
// Previous code information, including information that is enclosed by brackets
for (i = starting_year; i <= final year; i++) // loop is created from generated year to generated year
    {
    if ( i >= 1439 && i <= 1444) // if statement is created to account for years 1439-1944
        {
        continue; // loop stops if 1939 ≤ i ≤ 1944, then resumes.
        }
    else
        {
        System.out.println ("On October 13, " + i + ", there was a seminar devoted to the Battle of Hastings);
        }
// later code information
```

3-4. Let us find out the total attendance of these seminars. Write a piece of code that, give the starting and the ending years, asks the user for the number of attendees in each of these years, and adds them all together. We start with the total equal to 0, and then, for each year, add the number supplied by the user to this total. For example, if we had 10 attendees in the first year and 12 in the second year, we should first compute 0 + 10 = 10, then 10 + 12 = 22.

20
20

```
// Previous code information, including Scanner information
        Scanner input = new Scanner (System.in);
System.out.println ("Please enter the starting year:");
    int starting_year = input.nextInt(); // first year is generated
System.out.println ("Please enter the final year:");
    int final_year = input.nextInt(); // last year is generated
    int total = 0; // initial total is defined
for ( i = starting_year; i <= final_year; i++) // loop is created from first to last year
  {
    System.out.println ("How many people attended the seminar of year" + i);
        int attendees = input.nextInt (); // number of attendees per year is generated
        total = total + attendees; // total becomes new total plus sum of previous attendees
  }
System.out.println ("The total number of people who attended seminars
                    from the year" + starting_year +
                    " to " + final_year + " is " + total); // total is printed out
// remaining code information
```

5-6. It turns out that the number of seminar attendees grows quadratically with time: at year i, we had i^2 attendees. So, between years m and n, we have $m^2 + (m+1)^2 + \ldots + n^2$ attendees. Write a code that, for given integers m and n, computes this sum. Trace your code for m = 1 and n = 3.

20/20

```
import java.util.Scanner;
public class Seminar
{
public static void main(Args[] )
{
System.out.println ("Enter the starting year");
    int m = input.nextInt(); //first year is generated
System.out.println("Enter the ending year");
    int n = input.nextInt(); //last year is generated
int sum = 0; //sum with initial value 0 is defined
for (int i = m; i <= n; i++) // loop from m to n is created
{
    sum = sum + (i*i); // sum will be previous sum plus square of current i
}
System.out.println("The number of attendees in the seminar is " + sum);
} // Total Sum is printed
}
```

General Code for (generating) inputting numbers

Tracing for m=1 and n=3

```
public class Seminar
{
public static void main(Args[])
{
int m=1; //m=1 is defined
int n=3; n=3 is define
int sum=0;
for (int i=m; i<=n; i++) //loop is defined
{
sum=sum+(i*i);
}
System.out.println("The number of attendees in the seminar is "+sum);
}
}
```

At i=1, we have
sum = 0+(1*1)=1
and then i becomes 2, satisfies i<=n condition, so
sum = 1+(2*2)= 5;
i becomes 3, satisfies i<=n condition so
sum = 5+(3*3) = 14
i becomes 4, does not satisfy i<=n condition (loop stops)

Output is: "The number of attendees in the seminar is 14.

7. Trace, step-by-step, what will be the result of following Java code:

```
int n = 11;
String bin = "";
while(n > 0){
  if (n % 3 == 1)
    {bin = "1" + bin;}
  else if(n % 3 == 2)
    {bin = "1" + bin;}
  else
    {bin = "0" + bin;}
  n = n / 3;
}
{System.out.println(bin);}
```

10/10

Draw the boxes corresponding to all the variables, and show all the changes of their values.

Original values { 11 (n) ⇒ n>0 ✓ ⇒ loop takes place ⇒ 11%3 = 2, so first else statement takes place
∴ bin = "1" + bin = 1
n = 11/3 = 3
bin (String)

First Iteration values { 3 (n) ⇒ n>0 ✓ ⇒ loop takes place ⇒ 3%3 = 0, so second else statement occurs
1 (bin)
∴ bin = "0" + 1 = 01
n = 3/3 = 1

Second Iteration values { 1 (n) ⇒ n>0 ✓ loop takes place ⇒ 1%3 = 1, so first if statement takes place
01 (bin)
∴ bin = "1" + 01 = 101
n = 1/3 = 0

Third Iteration values { 0 (n) ⇒ n>0 ✗ Loop stops
101

Output is 101

8-9. A foot is defined as 12 inches, i.e., 12 * 2.54 = 30.48 cm. To help students transform feet to centimeters, write a main method that asks the user for a number n, and prints a table describing how many cm are in 1 ft, 2 ft, ..., all the way to n feet. For example, if the number is 3, your program should print the following table:

```
1 ft is 30.48 cm
2 ft is 60.96 cm
3 ft is 91.44 cm
```

20/20

```
import java.util.Scanner;
public class [illegible]Conversion
{
public static void main (Args[] ) // main method
{
System.out.println(" Enter your number of feet: "); // asks user for number
        int n = input.nextInt(); // number n is generated
int feet = 0; // feet number at 0 is defined
for (int i=1; i<=n; i++) // loop starting at 1 and ending at n is defined
{
    feet = feet + 30.48; // 30.48 cm are added for each iteration
  System.out.println(i + " ft is " + feet + " cm");
} // table is printed
} // end main method
}
```