Seneca College Winter 2020

Applied Arts & Technology

SCHOOL OF COMPUTER STUDIES

JAC444 Demo Due date : Jan 17, 2020 Final Code Submission Date: Jan 18, 2020

Workshop 1

Notes:

i. Each task should be presented during the lab, demo worth 70% of the workshop marks and code uploading worth the other 30%.

ii. All the tasks should be demoed in Jan 17th lab.

iii. Make sure you have all security and check measures in place, such as wrong data types etc., no need to implement Exception as we haven't covered yet. There are other ways to handle bad input data.

iv. Given output structure is just for student to have a glimpse what the output can look, students are free to make the output better in any way.

Other inputs can be given during demo, so make sure you test your program properly.

Task1. Zeller's congruence is an algorithm developed by Christian Zeller to calculate the day of the week. The formula is

$$h = \left(q + \frac{26(m+1)}{10} + k + \frac{k}{4} + \frac{j}{4} + 5j\right)\%7$$

where

· h is the day of the week (0: Saturday, 1: Sunday, 2: Monday, 3: Tuesday, 4:

Wednesday, 5: Thursday, 6: Friday).

· q is the day of the month.

· m is the month (3: March, 4: April, …, 12: December). January and February

are counted as months 13 and 14 of the previous year.

· j is the century (i.e., $year100$)

· k is the year of the century (i.e., year % 100).

Note that the division in the formula performs an integer division.

Write a program that prompts the user to enter a year, month, and day of the month, and displays the name of the day of the week. Here are some sample runs:

```
Enter year: (e.g., 2012): 2015 Enter
Enter month: 1-12:  1 Enter
Enter the day of the month: 1-31: 25 Enter
Day of the week is Sunday
```

```
Enter year: (e.g., 2012): 2012 Enter
Enter month: 1-12: 5 Enter
Enter the day of the month: 1-31: 12 Enter
Day of the week is Saturday
```

(Hint: January and February are counted as 13 and 14 in the formula, so you need to convert the user input 1 to 13 and 2 to 14 for the month and change the year to the previous year.)

Task2. The monthly payment for a given loan pays the principal and the interest. The monthly interest is computed by multiplying the monthly interest rate and the balance (the remaining principal). The principal paid for the month is therefore the monthly payment minus the monthly interest.

Write a program that lets the user enter the loan amount, number of years, and interest rate and displays the amortization schedule for the loan. Here is a sample run:

```
Loan Amount: 10000 Enter
Number of Years: 1 Enter
Annual Interest Rate: 7 Enter

Monthly Payment: 865.26
Total Payment: 10383.21

Payment#     Interest     Principal     Balance
1            58.33        806.93        9193.07
2            53.62        811.64        8381.43
...
11           10.0         855.26         860.27
12            5.01        860.25           0.01
```

Note: The balance after the last payment may not be zero. If so, the last payment should be the normal monthly payment plus the final balance.

Hint: Write a loop to display the table. Since the monthly payment is the same for each month, it should be computed before the loop. The balance is initially the loan amount. For each iteration in the loop, compute the interest and principal, and update the balance. The loop may look like this:

```
for (i = 1; i <= numberOfYears * 12; i++) {

interest = monthlyInterestRate * balance;

principal = monthlyPayment - interest;

balance = balance - principal;

System.out.println(i + "\t\t" + interest

+ "\t\t" + principal + "\t\t" + balance);

}
```

Task3. Write a program that plays the popular scissor–rock– paper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates a number 0, 1, or 2 representing scissor, rock, and paper. The program prompts the user to enter a number 0, 1, or 2 and displays a message indicating whether the user or the computer wins, loses, or draws. Here are sample runs: (Hint: Use switch statement)

```
scissor (0), rock (1), paper (2): 1 Enter
The computer is scissor. You are rock. You won
```

```
scissor (0), rock (1), paper (2): 2 Enter
The computer is paper. You are paper too. It is a draw
```