Program 3

CS2010 25 pts

Spring, 2018 Due: 11:59pm, Wed., Feb. 14, 2018

**Program description:** The Adventure Campus Rewards credit card offers cardholders the opportunity to earn points every time they use the card. Card holders earn 5 points for every $1 spent on entertainment (music, movies, video game rentals and sporting events), 2 points for every $1 spent on food (at restaurants, fast food places and coffee shops) and 1 point for every $1 spent on everything else. Cardholders can redeem their earned points for movie rentals (150 points each), pizzas (300 points each) or downloads from iTunes (60 points each). Create a complete C++ project in Visual Studio to find the number of points earned and the type and number of rewards selected by a cardholder for charges made on the card. Use your last name, first initial and pgm3 as the project folder and C++ file name (e.g., **ShortM\_pgm3** and **ShortM\_pgm3.cpp**).

**Input:** Prompt the user to enter the dollar amount of entertainment purchases, food purchases and all other purchases charged on the Adventure Campus Rewards credit card. Then ask the user to enter a one-letter code indicating the student's first choice for rewards: "M" for movie rentals, "P" for pizzas, "I" for iTunes downloads.

D300

**Processing:**

1. First find the total points earned by a cardholder based on the dollar amount of each kind of purchase. For example, if the user spent $100 on entertainment purchases, $60 on food purchases and $155 on all other purchases, he/she would have earned 775 points (5 \* $100 + 2 \* $60 + 1 \* $155).

2. Then look at the student's preference for rewards. If a "P" is entered, indicating that the cardholder's first choice for rewards is pizzas then determine how many pizzas he/she may get with the points earned. If there are any points left over, figure out how many movie rentals he/she may get with the remaining points. Figure the number of iTunes downloads with any remaining points.

For example, if the student earned 775 points and choose pizzas as the first choice for rewards, he/she could get 2 pizzas, 1 movie rental and 0 iTunes downloads with 25 points left over. If a student enters "M" for movie rentals as his/her first choice for rewards then figure the number of movie rentals he/she could get first (5 in this case) and the number of iTunes downloads (0 in this case) with any remaining points with 25 points left over. If a student enters "I" for iTunes as his/her preference for rewards then just figure the number of iTunes downloads he/she could get (12) with 55 points left over.

**Output:** Test your program with several different sets of values. Do the calculations by hand and compare them to your program results to make sure that your program is working correctly. Here is an example of a program run showing the input and output.

Adventure Campus Rewards Program

Enter dollar amount of your Entertainment purchases: $100

Enter dollar amount of your Food purchases: $60

Enter dollar amount of any other purchases: $155

Total number of points earned: 775

Enter your first choice for rewards-(M)ovies, (P)izzas or (I)tunes: P

You have earned the following rewards for your credit card purchases!

2 Pizzas (300 points each)

1 Movie rentals (150 points each)

0 iTunes downloads (60 points each)

25 points leftover

**Program Documentation & Style:**

1. Declare all constants and variables that your program uses at the beginning of your program.

2. Your program should include two types of comments. BE SPECIFIC!

a. Header comments at the beginning of your program including lines with:

- Your name, course name, and class time

- Program assignment number, program file name (pgm3.cpp) and due date

- **Purpose**: a sentence or two explaining the purpose of the program

- **Input**: a description of the input data needed by the program when you run it

- **Processing**: a description of the processing (calculations) done by the program

- **Output**: a description of the results (output) produced by the program

b. In-line comments: There should be

an in-line comment for each main step in your program. In general, this means at least one comment with each group of C++ statements that handles the input, the processing and the output steps of your program.

3. Use meaningful identifier names

4. Include clear prompts for the user about entering the data.

5. Include clear descriptions of the results when you display them.

6. Format your output neatly.

### Turning in your program

When you are satisfied that your program is producing the correct output, close your project (File 🡪Close Solution or quit Visual Studio), zip the entire project folder (right-click on the project folder and choose Send To 🡪 Compressed (zipped) folder) and upload the zipped file on Canvas.

### Grading Rubric

**Points**

5 Header comments complete, In-line comments complete

2 Uses good program style (meaningful data names, white space, indentation)

3 Accepts input values, includes clear prompts

4 Displays error message if user enters invalid letter for rewards choice

8 Calculated fields correct

3 Output neatly labeled and displayed

**25 Total Points**