PROGRAM 5

Design Due Date **April 19 by 11:59PM**

Program Implementation Due Date: **April 27 by 11:59PM**

You are to complete the following on your own unless otherwise stated. All programs must be written in C++ and be Linux compatible. Make sure you follow the ***Assignment Guidelines*** and ***Program Guidelines*** posted on my course webpage.

Design due date deliverables – use the *Design Template*

- Complete the identifier dictionary, structure chart & algorithms (no test plan)
    - You may draw the structure chart on a white sheet of paper. Using a ruler and your best penmanship, complete the structure chart on the piece of paper. When complete you can scan or take a picture of your drawing. Be sure the picture is CLEAR. This picture can be inserted on the structure chart page of the design document.
- Submit the design document to "Program 5 Design" dropbox on D2L

Implementation due date deliverables:

- Copy program source code into **p5_cpp.txt** and upload to "Program 5 Implementation" dropbox on D2L
- Concatenate p5 report files to **p5_rpt.txt** and upload to "Program 5 Implementation" dropbox on D2L
    - to concatenate your files into one file execute the following on the command line:

```
cat p5run1.rpt p5run2.rpt> p5_rpt.txt
```

- Concatenate p5 output files into **p5_out.txt** and upload to "Program 5 Implementation" dropbox on D2L
    - to concatenate your files into one file execute the following on the command line:

```
cat p5run1.out p5run2.out> p5_out.txt
```

- Complete Written log and summary (use Work log & summary Template) and upload to "Program 5 Work Log and Summary".

<u>Design, test and implement the following C++ program:</u>
Make any necessary correction to program 4 before you begin program 5.

All numeric input should be valid. Your program should trap the end-user at the input until the user enters a valid value. You need to check for data value being in bounds and the data type entered (by checking stream failure). No integer value should be less than 1. The child's age should be less than 120 and the number of guests should be less than 500.

The program should be set up to process a number of birthday parties. After each party is processed ask the user if they have another party to process. Keep count of the number of parties and the total amount spent on all parties combined.

Each party should be processed just as it was in program 4, using the same pricing and report format. All output will be written to a single output file. Ask the user for the input filename just once at the start of the program. Do not keep the output file open the entire program. Open and close the file using append mode each time you write a birthday party report.

When the user is done entering birthday parties the program should output a summary report that includes the number of parties, total cost of all parties and the average being spent on each party.

<u>Implementation Requirement</u>:
All input, output, and calculations should be implemented in programmer-defined functions. The main should not have any detail code in it – no input, output or assignment statements. There can be control statements (selection and loops) and function calls in the main function. There must be at least four (4) function calls in the main function. Each function should do one task (can be a generalized task).

Print 2 lines of asterisks followed by 2 blank lines at the end of the summary report. Print the same lines to the screen at the end of the program.

Implementation changes to transition from program 4 to program 5

Each bullet identifies a revision that should be fully tested BEFORE moving to the next bullet.

- Correct any errors in p4.cpp
- Copy p4.cpp into p5.cpp and make the remaining changes below to p5.cpp
- Add an "endl" after each user prompt. The reason for this will be explained in class.
  Example:
  ```
  cout << "What is the name of the birthday child: " << endl;
  ```

- Change the output lines at the top of the **report** describing birthday child, age and guests invited. See the example report file output for program 5.
- Move each group of calculations (ex: calculations relating to balloons) into a programmer defined function. Use pass by reference if there is more than one value being "returned" (ex: number of balloons, cost of balloons).
- Add validation for all numeric input. This can be done by adding a parameterized general function that validates an integer or can be placed in EACH of the input functions dealing with an integer input value.
- Move the call to get the filename to main( ) and get this value FIRST from the user.
- Add a counter for number of parties and an accumulator for total cost of all parties. Determine where to initialize these and where they should be updated. It is permitted to update these in the main( ).
- Add a function to print the summary report and calculate the average spent on each party (right now there is just ONE party).
- Add the loop to main( ) to process the multiple parties (determine the function calls that need to be inside the loop, repeated for each party, and those that should execute ONE time either at the start or end of the program.)
- So that is will be easier to read, Please be sure to print 2 lines of asterisks followed by 2 blank lines at the end of the summary report and print the same to the screen at the end of the program.

Input Data and Redirection

Because there is so much input I have *canned* the input data into two files for two runs. This technique of creating files with user input values is used for testing programs that have a lot of user input. The more user input there is, the hard it is to test the program accurately. Your program will think the input is coming from the keyboard, but instead when you run the program you will redirect the *cin* statements to extract/get the values from the file. This redirection is done from the Linux command line when you run your program. There is NO CHANGE to the C++ code. The difference will be where the characters will be extracted from in each of the "cin statements".

So that I can see your prompts and your error messages (they should match the ones shown here in the specifications) you will also redirect the cout statements into a file. This will be done instead of creating a script file. To capture the prompts and other output that is inserted to the monitor (all *cout* statements) you should redirect the output on the Linux command line. To make these redirected output files more readable, please put an endl at the end of each user prompt in your C++ program. This is the only change to made to your C++ program because you will be using redirection of your output prompts.

The report file that is created by your C++ program is separate from the files used for redirection. NEVER use the same names for files used for redirection and those used in your C++ program!!

Example of a run using redirection of both input and output
(this is done on Linux command line)

```
p5.exe < p5run1.in > p5run1.out
```

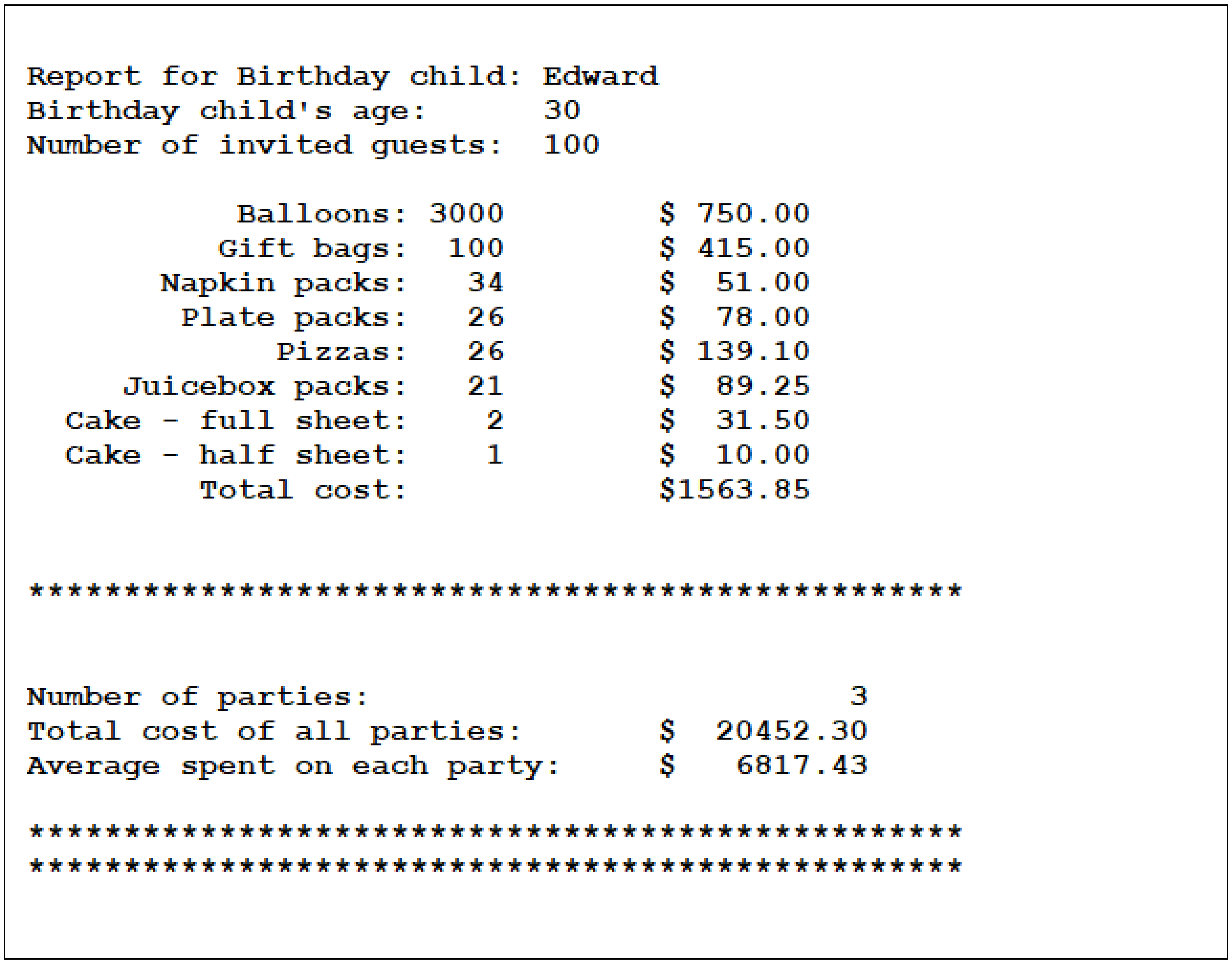
This runs *p5.exe* using the input file *p5run1.in* instead of entering the values from the keyboard. It also creates an output file called *p5run1.out* and writes all data into that file that would have been written to the monitor (cout).

The canned input files for redirection can be copied from my directory:
~zimmer/csci130/202/p5/p5run1.in (only valid data)
~zimmer/csci130/202/p5/p5run2.in (both valid and invalid data)

Example Run (this is what p5run2.out should look like. Notice that the actually values provided from p5run2.in are not showing up in p5run2.out)

```
Welcome to the Birthday Party Cost Calculator

Please enter the file name to use for the report:

What is the name of the birthday child:
How old will the birthday child be?
ERROR, Please enter a value in-range!

How old will the birthday child be?
ERROR, Please enter an integer!

How old will the birthday child be?
ERROR, Please enter a value in-range!

How old will the birthday child be?
How many invited guests?
ERROR, Please enter a value in-range!

How many invited guests?
ERROR, Please enter a value in-range!

How many invited guests?
ERROR, Please enter an integer!

How many invited guests?
Would you like to process another party (Y/N)?


What is the name of the birthday child:
How old will the birthday child be?
How many invited guests?
Would you like to process another party (Y/N)?


What is the name of the birthday child:
How old will the birthday child be?
How many invited guests?
Would you like to process another party (Y/N)?


*************************************************
*************************************************
```

<u>Example Report and Summary Report</u> (this is the end report and summary for p5run2.rpt)



```
Report for Birthday child: Edward
Birthday child's age:      30
Number of invited guests:  100

           Balloons: 3000        $ 750.00
          Gift bags:  100        $ 415.00
       Napkin packs:   34        $  51.00
        Plate packs:   26        $  78.00
             Pizzas:   26        $ 139.10
     Juicebox packs:   21        $  89.25
  Cake - full sheet:    2        $  31.50
  Cake - half sheet:    1        $  10.00
         Total cost:             $1563.85


*************************************************


Number of parties:                          3
Total cost of all parties:       $  20452.30
Average spent on each party:     $   6817.43

*************************************************
*************************************************
```