Project 2
The Amusement Park Problem – Part II

Requirements

First correct any errors noted by instructor in Part I.

Now change the pricing in the Part I program to include the following price changes and adjustments.

Category	Price	Adjustment
Children	$12.00	For every 4 children, one is free. This means that for every 4 tickets purchased, the 4th ticket is free. *(This is a one statement calculation.)*
Adult	$26.50	The adult cost (not the price) is reduced by $3 if there are more than 5 adults.

Add the total number of tickets to the display.

All groups with more than 20 people or 14 or more children must pay a $15.00 security fee. Add the security fee to the total bill. Only display the security fee if it's not zero..

Add a pretest while loop so the program executes until the user enters **-1** for the child ticket value. See the pretest loop example in the Chapter Two *Repetition Notes*. Review the example that uses **price** to determine how the loop executes.

Add a posttest do loop to verify that the cash entered is enough to pay the bill. If it is not, display a message and request the cash again.

Add a confirmation number to the receipt. The first number should be 100 and then add 1 for each new receipt.

Here is a sample display:

```
               Chesapeake Amusement Park

Enter children tickets or -1 to stop... 8
Enter adult tickets.................... 6

               Chesapeake Amusement Park
               -------------------------

                Tickets          Price     Total
        Child       8            12.00     72.00
        Adult       6            26.50    156.00

                   14

             Total Bill                $    228.00

             Cash received......200
           Cash must be >= Total Bill

             Cash received......230

          Change                              2.00
          Confirmation number = 100

Enter children tickets or -1 to stop...
```

Purpose of this project

Develop C++ program with the following new features:

- if statements
- Pretest while loop
- Posttest do loop

Advice from the Instructor:

- Make one change at a time and test the change before continuing.
    - Add the while loop first to make testing easier.
    - Change the child pricing.
    - Change the adult pricing.
    - Add the ticket total.
    - Add the confirmation number.
    - Check for the security fee.
    - Add the do loop to check cash received.

- Problems that students have with this project:
    - Continue not to set up constants.
    - Inefficient if statements. If you use the same statements in the true path as you do in the false path, you should probably place the redundant code once before or after the *if* statement.
    - Don't incorporate the *Class Standards*.

Single-entry/single-exit

In the *Class Standards* I talk about single-entry/single-exit. It is the preferred method for coding ifs and loops. Here is the text:

There is a concept in programming called single-entry/single-exit. The path for any loop or if structure should enter at the same place and exit at the same place. Do NOT use a break or continue statement to jump out a loop.

I know there is more than one right way to code a solution, but I want you to use the pretest while loop as I have explained in the examples so you know how to code this way if asked in future classes or jobs. Here is a very rough draft of the pseudocode for Project Two:

get children tickets
while(*children tickets* != -1)
//start of loop
    get adult tickets
    calculations
    displays
    get children tickets
//end of loop

With a pretest while loop, you get the sentinel value (*children tickets*) before the while and at the end of the loop.

CSIT 210 *Introduction to Programming* (Java) is a prerequisite for this course. You will find that C++ statements are very similar to Java so you can rely on your Java skills in this class. The selection and repetition statements are the same in C++ as they were in Java.