

Introduction To Computer Programming C++

Mr. Clausen

Program C6A

Program 6A mp3 Player Stats 30 points

We will write a program that practices using a `getline` statement after a `cin` statement, uses `setw`, `setf`, special characters, and `setprecision` while gathering some information about your mp3 player. **If you don't own an mp3 player, feel free to make up the information as you use the program.** Save the program as `LastNameFirstNameP6A.cpp` in your "S:" directory. To see a model for this program, look at the source codes for the programs "books.cpp" and "pizza.cpp" in the network directory titled: `IntroCompProgFiles`. Look in the folder, `Other C++ Resources` and the folder `Other Textbook Examples`.

As you type all your programs this year, be sure not to type past the 80-column line in Borland C++ 5.02 for Windows. If you have any statements longer than 80 columns, press the return key to "wrap" the statement around to the next line.

- 1) Type comments at the beginning of the program to display your name and other information just like those used for program 1A. **Make sure to change the program name and program description in these comments, so that the program number, name, and description say what is listed above.**
- 2) Include `<iostream.h>` (so you can use the `cout` and `cin` commands), and include `<conio.h>` so you can use `getch()` to leave your output displayed on the screen until the user presses any key to continue, and so you can use `clrscr()` to clear the screen. Also `#include <iomanip.h>` so you can use `setw` and `setprecision`. You will need to also type: `#include "ostring.cpp"`; (The textbook says to use `ostring.h`, but Borland will work with the `cpp` file and we will not need to create a project for this to work.)
- 3) Declare a constant of type `double` named `PRICE_PER_SONG` and set it equal to 0.99 (99 cents).
- 4) Inside the `int main()` function, on the first line below the left curly bracket that begins the main function, declare variables of type `int` for `age` and `number_of_songs`. Declare variables of type `double` for `mp3_player_cost`, `total_song_cost`, `total_cost`, and `average_cost_per_year`. Declare variables (instances) of type `ostring` for `full_name`, `mp3_player_name`, `consume_newline`, and `favorite_song`. In these declarations, initialize the integers to 0 (zero), and the variables of type `double` to 0.0. Initialize `full_name`, `mp3_player_name`, `consume_newline`, and `favorite_song` to "" which is considered to be the null string (two double quotes with nothing in between). You will need the `consume_newline` variable to "flush the input stream" whenever you use a `getline` statement after a `cin` statement. **This is the preferred method rather than the method that our textbook advocates, therefore, please use `getline (cin, consume_newline);` rather than `cin.ignore (80, '\n');` to flush the input stream.**
- 5) In all our programs, follow the Input, Calculations, and Output organization of your program. Make sure that you include the following comment lines in the `int main ()` portion of your program (each comment followed by the appropriate source code).
- 6) After the variable declarations (before the input section) use `cout` statements to display your name and period output just like those used for program 1A **Make sure to change the program name and program**

description in these cout statements. Start these commands with the following statement:

```
//-----Display My Information-----
```

7) For the Input section, **ask the user to enter the following information in the order that I am listing here. You will lose points if you change the order of the input statements.** Make sure to have a user friendly prompt for each question. Ask the user for an integer that represents their age, then ask for their full name (first and last names). Next ask how many songs are on their mp3 player, and then ask them for the name of their mp3 player (This should be a name that is more than one word. If it is an iPod then specify if it's an iPod nano, iPod 30Gb, iPod Shuffle, iPod Video, etc. For other types of mp3 players type more than one word also, for example, Creative Zen Ultra, Rio Carbon, iRiver H10, etc.). Then ask the user how much their mp3 player cost (don't worry about tax or extended warranties). Last, ask the user the name of their favorite song that is on their mp3 player. Make your program user friendly by prompting them for each of these values. Start these commands with the following statement:

```
//-----Input-----
```

After all of the information is input, tell the user to press any key to continue, have a getch(); command and then use a clrscr() command to clear the screen before doing the calculations and output sections of the program. You will need the consume_newline variable to "flush the input stream" whenever you use a getline statement after a cin statement. **This is the preferred method rather than the method that our textbook advocates, therefore, please use getline (cin, consume_newline); rather than cin.ignore (80, '\n'); to flush the input stream. For this program, I would like you to use endl instead of '\n' to practice the other way to use the end line command.**

8) The calculations section of your program should consist of the comment line and the commands to perform the calculations listed below. Start these commands with the following statement:

```
//-----Calculations-----
```

```
total_song_cost = number_of_songs * PRICE_PER_SONG ;  
total_cost = total_song_cost + mp3_player_cost;  
average_cost_per_year = total_cost / double (age); //Remember type casting? Don't type this comment...
```

9) For the output section of the program display the output as shown below. For your output, use the setf commands for fixed, showpoint, and setprecision for 2 decimal places since we are using dollars and cents. In order to show the apostrophe when listing the mp3 owner's name, use the special character '\'. To place quotes around the name of the favorite song, use the special character '\" at the beginning and ending of the song name. Use the setw command to allow 10 places for the user's age, and 10 places for the number of songs. Start these commands with the following statement:

```
//-----Output-----
```

Your output should be formatted to look like the following:

```
Name: Steve Jobs      Age: 50      Number of Songs: 534  
mp3 player: iPod Video 60Gb  
mp3 player cost: $399.99  
Total Cost For Songs: $528.66  
Total for Player and Songs: $928.65  
Average Cost Per Year: $18.57
```

Steve Jobs' favorite song is: "Money Grows on the Apple Tree "

When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don't need to make any changes, then save your program in the "W" network mapping, and the Program 6A folder.