

```

/
*****
*****
PROGRAM : Blood_Alcohol_Concentration_Chart_creator.cpp
PURPOSE : This program will create a nice BAC [Blood Alcohol Concentration] Chart for both Male
and Female persons, which is
    strongly correlated with driving impairment. For this reason, the program will take two integer
values, [one for
    getting the weight of the user, another for having the time(in minutes) since the user had his/her
last drink], and a character
    value [for getting the information, whether the user is 'M' for male or 'F' for female.] as inputs.
After giving all these
    information properly as inputs, the program will show a whole detailed BAC chart [for that
particular inputs.] as an output.
Coder : Mosfiqur Rahman (mr986@drexel.edu)
Last Modified : 16th February, 2016.
*****
*****/

```

```

#include <iostream> // for having input and output
#include<iomanip> // for formatting output

using namespace std; // for using standard library namespace

void computeBloodAlcoholConcentration(int numDrinks, int weight, int duration, double &maleBAC,
double &femaleBAC);
// prototyping function computeBloodAlcoholConcentration, which will compute the value of BAC for
both males and females
// under certain number of drinks, weight, and duration.
string impairment(double bac);
// prototyping function impairment, which return different driving impairments in strings under
different values of BAC.
int promptForInteger(string const &message, int lower, int upper);
// prototyping function promptForInteger, which is used to prompt a directed message, and to get the
weight, and
// the time[since the user had his/her last drink] as inputs
char promptForMorF(string const &message);
// prototyping function promptForMorF, which is used to prompt a directed message, and to get the
gender[M or F] of the user.
void showImpairmentChart(int weight, int duration, bool isMale);
// prototyping function showImpairmentChart, which will generate the final output[i.e. the entire BAC
chart] calling the
//computeBloodAlcoholConcentration(int numDrinks, int weight, int duration, double &maleBAC,
double &femaleBAC)
// and impairment(double bac) function in it

const double safe = 0.00;
// declaring, and initializing a global constant double variable safe
const double someImpairment = 0.04;

```

```

// declaring, and initializing a global constant double variable someImpairment
const double significantAffected = 0.08;
// declaring, and initializing a global constant double variable significantAffected
const double someCriminalPenalties = 0.10;
// declaring, and initializing a global constant double variable someCriminalPenalties
const double deathPossible = 0.30;
// declaring, and initializing a global constant double variable deathPossible
const string SAFE = "Safe To Drive";
// declaring, and initializing a global constant string variable SAFE
const string SOMEIMPAIR = "Some Impairment";
// declaring, and initializing a global constant string variable SOMEIMPAIR
const string SIGNIFICANT = "Driving Skills Significantly Affected";
// declaring, and initializing a global constant string variable SIGNIFICANT
const string MOST_STATES = "Criminal Penalties in Most US States";
// declaring, and initializing a global constant string variable MOST_STATES
const string ALL_STATES = "Legally Intoxicated - Criminal Penalties in All US States";
// declaring, and initializing a global constant string variable ALL_STATES
const string YOURE_DEAD = "Death is Possible!";
// declaring, and initializing a global constant string variable YOURE_DEAD


//The following function is the main function
int main()
{
    long victim_weight, drink_duration;
    // declaring two long integer variable to store the value of the time [since the user had his/her last
drink]
    // and the weight of the user
    char gender;
    // declaring a character variable to store the gender of the user as either 'M' for males, and 'F' for
females
    bool isMale;
    // declaring a bool variable, which will store particular value based on the gender of the user
    cout << "Welcome to BAC [Blood Alcohol Concentration] Chart creator!" << endl << endl;
    // Optional, introductory texts

    victim_weight = promptForInteger("Please, enter your weight [in pounds]: ", 0, 1000000);
    // Calling the promptForInteger function, passing the reference message, and the value for the
variable lower and upper
    // Storing the value in the variable victim_weight, which is calculated by the promptForInteger
function with the passed value

    drink_duration = promptForInteger("Great! Now, please, enter the time [in minutes] since you last
drink: ", 0, 1000000000);
    // Calling the promptForInteger function, passing the reference message, and the value for the
variable lower and upper
    // Storing the value in the variable drink_duration, which is calculated by the promptForInteger
function with the passed value

```

```

gender = promptForMorF("At last, please tell me your gender [as 'M' or 'F']: ");
// Calling the promptForMorF function, passing the reference message
// Storing the value in the variable gender, which is calculated by the promptForInteger function with
the passed value

```

```

if(gender == 'M')
// Checking whether the value of the variable gender is 'M' or not
//If it's so then continues, or moves to other conditional statement.
{
isMale = 1;
// Initializing the value for the bool variable isMale
}
else
// If the previous 'if' statement fails, then the program runs through this statement.
{
isMale = 0;
// Initializing the value for the bool variable isMale
}

showImpairmentChart(victim_weight, drink_duration, isMale);
// Calling the showImpairmentChart function, and passing the value of weight, duration, and isMale
// through the variables victim_weight, drink_duration, isMale respectively
return 0;
}

```

```

/****

```

The following function will compute the BAC [Blood Alcohol Concentration] for both males, and females.

```

@param numdrinks - Number of drinks taken by the user.
@param weight    - Weight [in pounds] of the user
@param duration  - Time [in minutes] since the user had his/her last drink
@param &maleBAC  - The value of BAC [Blood Alcohol Concentration] of a male person
@param &femaleBAC - The value of BAC [Blood Alcohol Concentration] of a female person

```

```

****/

```

```

void computeBloodAlcoholConcentration(int numDrinks, int weight, int duration, double &maleBAC,
double &femaleBAC)

```

```

// Defining the computeBloodAlcoholConcentration function, which will compute the value of BAC
for both males and females

```

```

{
const double MALE_CONSTANT = 3.8;
//Declaring and initializing a constant double variable for MALE_CONSTANT
const double FEMALE_CONSTANT = 4.5;
//Declaring and initializing a constant double variable for FEMALE_CONSTANT

```

```

    maleBAC = MALE_CONSTANT * numDrinks / weight - ( .01 / 40 ) * duration;

```

```

// Calculating the BAC for a male person through the following algorithm
// Step: 1- First multiplying the value of MALE_CONSTANT with the value of [number of
drinks / the weight of the user]
// Step: 2- Then subtracting the value of [( .01 / 40 ) * duration since the user had his last drink]
//          from the previous value to get the final value

```

```

if(maleBAC < 0)
// Checking whether the final value of maleBAC is less than 0 or not
//If it's so then continues, or moves to other conditional statement.
{
    maleBAC = 0;
    // Initializing the value of maleBAC to 0
}

```

```

femaleBAC = FEMALE_CONSTANT * numDrinks/ weight - (.01 / 40) * duration;
// Calculating the BAC for a female person through the following algorithm
// Step: 1- First multiplying the value of FEMALE_CONSTANT with the value of [number of
drinks / the weight of the user]
// Step: 2- Then subtracting the value of [( .01 / 40 ) * duration since the user had his last drink]
//          from the previous value to get the final value

```

```

if(femaleBAC < 0)
// Checking whether the final value of femaleBAC is less than 0 or not
//If it's so then continues, or moves to other conditional statement.
{
    femaleBAC = 0;
    // Initializing the value of femaleBAC to 0
}

```

```

}

```

```

/****

```

The following function will return different driving impairments in strings under different values of BAC.

@param bac - The value of BAC [Blood Alcohol Concentration] of the user  
 @return bac\_condition\_brief - Driving Impairments in a string under the certain value of BAC of the user

```

****/

```

```

string impairment(double bac)
// Defining the impairment function, which will return different driving impairments in strings under
different values of BAC
{
    string bac_condition_brief;
    // Declaring a string variable bac_condition_brief to store the Driving Impairments

```

```

if(bac == safe)
// Checking whether the value of bac is equal to the value of the global constant safe
//If it's so then continues, or moves to other conditional statement.
{
    bac_condition_brief = SAFE;
    //Initializing the value of bac_condition_brief according to the value of the global constant SAFE
}
else if(bac > safe && bac < someImpairment)
// Checking whether the value of bac is greater than the value of the global constant safe
//and less than the value of the global constant someImpairment
//If it's so then continues, or moves to other conditional statement.
{
    bac_condition_brief = SOMEIMPAIR;
    //Initializing the value of bac_condition_brief according to the value of the global constant
SOMEIMPAIR
}
else if(bac >= someImpairment && bac < significantAffected )
// Checking whether the value of bac is greater than or equal to the value of the global constant
someImpairment
//and less than the value of the global constant significantAffected
//If it's so then continues, or moves to other conditional statement.
{
    bac_condition_brief = SIGNIFICANT;
    //Initializing the value of bac_condition_brief according to the value of the global constant
SIGNIFICANT
}
else if(bac >= significantAffected && bac < someCriminalPenalties)
// Checking whether the value of bac is greater than or equal to the value of the global constant
significantAffected
//and less than the value of the global constant someCriminalPenalties
//If it's so then continues, or moves to other conditional statement.
{
    bac_condition_brief = MOST_STATES;
    //Initializing the value of bac_condition_brief according to the value of the global constant
MOST_STATES
}
else if(bac > deathPossible)
// Checking whether the value of bac is greater than the value of the global constant deathPossible
//If it's so then continues, or moves to other conditional statement.
{
    bac_condition_brief = YOURE_DEAD ;
    //Initializing the value of bac_condition_brief according to the value of the global constant
YOURE_DEAD
}
else
// If the previous 'if' and 'else if' statements fail, then the program run through this statement.
{
    bac_condition_brief = ALL_STATES;
    //Initializing the value of bac_condition_brief according to the value of the global constant

```

```
ALL_STATES
```

```
    }  
    return bac_condition_brief;  
    // returns the value of the variable bac_condition_brief  
}
```

```
/**
```

```
    The following function will be used to prompt a passed message, and to get the weight and  
the time[since the user had his/her last drink] as inputs
```

```
@param &message - Message to be prompted
```

```
@param lower    - Lower limit of the integer input
```

```
@param upper    - Upper Limit of the integer input
```

```
@return integer - The final value of the given input as an integer
```

```
***/  
  

```

```
int promptForInteger(string const &message, int lower, int upper)
```

```
// Defining the promptForInteger function, which will be used to prompt a passed message, and to get  
the gender['M' for male or 'F' for female] of the user
```

```
{  
    int integer;  
    // Declaring a integer type variable to store the integer input from the user
```

```
    do
```

```
    // A do-while loop to ask the user for a valid integer input
```

```
    // The loop will run repeatedly untill it gets a perfect input
```

```
    {  
        cout << message;  
        // Showing the passed message as an output message  
        cin >> integer;  
        // For getting the input value for the variable integer from the user
```

```
        if (integer >= lower && integer <= upper)
```

```
        // Checking whether the value of the variable integer is greater than or equal to the passed value  
of the variable lower
```

```
        //and less than or equal to the passed value of the variable upper
```

```
        //If it's so then continues, or runs the loop again
```

```
        return integer;  
        // returns the final value of the variable integer
```

```
    } while (!(integer >= lower && integer <= upper)); // Conditon to run the loop again
```

```
}
```

```
/**
```

The following function will be used prompt a passed message, and to get the gender['M' for male or 'F' for female] of the user.

@param &message - Message to be prompted

@return gender - The final value of the user's gender [i.e. 'M' for male or 'F' for female]

```
*/
```

```
char promptForMorF(string const &message)
```

```
// Defining the promptForMorF function, which will be used prompt a passed message,
```

```
//and to get the gender['M' for male or 'F' for female] of the user.
```

```
{
```

```
    char gender;
```

```
    // Declaring a character variable gender to store the information whether the user is
```

```
    // Male as 'M' or Female as 'F'
```

```
    do
```

```
    // A do-while loop to ask the user for a valid character input
```

```
    // The loop will run repeatedly until it gets a perfect input
```

```
    {
```

```
        cout << message;
```

```
        // Showing the passed message as an output message
```

```
        cin >> gender;
```

```
        // For getting the input value for the variable gender from the user
```

```
        if (gender == 'M')
```

```
        //Checking whether the value of the variable gender is equal to 'M' or not
```

```
        //If it's so then continues, or moves to other conditional statement.
```

```
        return gender;
```

```
        // returns the value of the variable gender
```

```
        else if (gender == 'F')
```

```
        //Checking whether the value of the variable gender is equal to 'F' or not
```

```
        //If both the previous 'if', and this 'else if' statement fails, then runs the loop again
```

```
        return gender;
```

```
        // returns the value of the variable gender
```

```
    } while (!(gender == 'M' || gender == 'F'));// Condition to run the loop again
```

```
}
```

```
/**
```

```
    The following function will generate the final output[i.e. the entire BAC chart] calling  
    the computeBloodAlcoholConcentration(int numDrinks, int weight, int duration, double  
&maleBAC, double &femaleBAC) and impairment(double bac) function in it
```

```
@param weight - Weight [in pounds] of the user
```

```
@param duration - Time [in minutes] since the user had his/her last drink
```

```
@param isMale - Particular value based on the gender of the user.[To be more exact- if male isMale =  
1, otherwise, isMale = 0]
```

```
***/  
  

```

```
void showImpairmentChart(int weight, int duration, bool isMale)
```

```
// Defining the showImpairmentChart function, which will generate the final output[i.e. the entire BAC  
chart]
```

```
{
```

```
    int number_of_drinks;
```

```
    // Declaring an integer variable to store the number of drinks taken by the user under some time  
constraint
```

```
    string gender;
```

```
    // Declaring a string variable to store the gender of the user in whole words with special output  
format
```

```
    if( isMale == 1)
```

```
    // Checking whether the value of the variable isMale is '1' or not
```

```
    //If it's so then continues, or moves to other conditional statement.
```

```
    {
```

```
        gender = "male, ";
```

```
        // Initializing the string value of gender
```

```
    }
```

```
    else
```

```
    // If the previous 'if' statement fails, then the program runs through this statement.
```

```
    {
```

```
        gender = "female, ";
```

```
        // Initializing the string value of gender
```

```
    }
```

```
    cout << endl << weight << " pounds, " << gender << duration << " minutes since last drink" << endl;
```

```
    // Showing the weight, and the duration since the has user has taken his/her last drink as output
```

```
    cout << "# " << "drinks" << "   BAC  Status" << endl;
```

```
    // Formatted Output to show the BAC status for different number of drinks under this line
```

```
    for(number_of_drinks = 0; number_of_drinks <= 11; number_of_drinks++)
```

```
    // Running a for loop to print the rest of the parts of the BAC [Blood Alcohol Concentration] Chart
```

```
    {
```

```
        double BAC_Final, maleBAC, femaleBAC;
```



```
// Declaring three different double variable to store and generalize the BAC value particularly for males and females
```

```
computeBloodAlcoholConcentration(number_of_drinks, weight, duration, maleBAC, femaleBAC);  
// Calling the computeBloodAlcoholConcentration function to compute the and generalize the BAC value
```

```
if(isMale == 1)  
    // Checking whether the value of the variable isMale is '1' or not  
    // If it's so then continues, or moves to other conditional statement.  
    {  
        BAC_Final = maleBAC;  
        // Initializing the value of the variable BAC_Final to maleBAC  
    }  
else  
    // If the previous 'if' statement fails, then the program runs through this statement.  
    {  
        BAC_Final = femaleBAC;  
        // Initializing the value of the variable BAC_Final to femaleBAC  
    }
```

```
cout << setw(8) << number_of_drinks << " " << fixed << setprecision(3) << BAC_Final << " " <<  
impairment(BAC_Final) << endl;  
    // Finally, Formatted Output to show the number of drinks and the BAC status for those different  
    number of drinks on the same line.  
    }  
}
```