

Jacob Wilkin
CSCE 303; OOP in C++
Small Project #2

Application Description: Implementing two file classes that have 3 global arrays with 3 corresponding global functions.

START APPLICATION

START main driver file

include iostream library
include Person.h class
include Friend.h class
Including namespace std

Start main function

```
Person agent_a ("Andrea", "Adams", Person movie, Person music, Person game)
Person agent_b ("Bernie", "Bates", Person movie, Person music, Person game)
Person agent_c ("Charlotte", "Caylorlie", Person movie, Person music, Person game)
Person agent_d ("Duayne", "Dramtoix", Person movie, Person music, Person game)
Person agent_e ("Edward", "Edmunds", Person movie, Person music, Person game)

Person agent_f ("Francous", "Frenchie")
Person agent_g ("Gerald", "Gramatix")
Person agent_h ("Hope", "Hjung")
Person agent_i ("Ivan", "Issiac")
Person agent_j ("Jessica", "Jones")
```

Declare array {Person, 5} // 5 for the attributes, first name, last name, movie, music, and game
People list = { List of agents from second constructor } //Agents that have only names listed.

```
Friends my_friends (people_list)
Update my_friends
Print favorites from my_friends
Print "check games"
```

End main function

END main driver file

START Person.h file

```
include iostream  
include namespace std
```

Start Person class

Start public

Friend class Friends

//Building constructors

Person (constant string, constant string)

Person (constant string, constant string, string, string, string)

//Destructor

~Person ()

string get_first_name ()

string get_last_name ()

string get_movie ()

string get_music ()

string get_game ()

void to_string ()

void set_movie (string)

void set_music (string)

void set_game (string)

//Creation of the 3 global functions

(Global) string select_music ()

(Global) string select_movie ()

(Global) string select_game ()

End public

Start private

```
    string first_name
    string last_name

    //Set none for the 3 global values
    String movie
    string music
    string game

    constant (Global) string music_genre [5] //iteration values
    constant (Global) string movie_genre [4]
    constant (Global) string game_genre [4]
```

End private

End Person class

END Person.h file

START Person.cpp file

```
Include file Person.h
include iostream library
include string library
include random library
including namespace std

constant string Person music_genre [ 5 ] {"world", "rock", "hiphop", "jazz", "classical"}
constant string Person movie_genre [ 4 ] {"documentary", "action", "adventure", "romance"}
constant string Person game_genre [ 4 ] {"adventure", "role play", "educational", "sports"}

string Person select_music()
    int music = rand() % 5 //randomize the 5 iterations
    return music_genre [ music ]

string Person select_movie()
    int movie = rand() % 4 //randomize the 4 iterations
    return movie_genre [ movie ]
```

```
string Person select_game()  
    int game = rand() % 4 //randomize the 4 iterations  
    return game_genre [ game ]
```

//Constructors and Destructor

```
Person Person(constant string the_first, constant string the_last)  
    : first_name { the_first }, last_name { the_last }  
    {}
```

```
Person Person(constant string first, constant string last, string movi, string musi, string gam)  
    : first_name {first}, last_name {last}, movie {movi}, music {musi}, game {gam}  
    {}
```

```
Person ~Person ( ) { }
```

//getters

```
string Person get_first_name ( )  
    return first_name;
```

```
string Person get_last_name ( )  
    return last_name;
```

```
string Person get_movie ( )  
    return movie;
```

```
string Person get_music ( )  
    return music;
```

```
string Person get_game ( )  
    return game;
```

//setters

```
void Person set_game (string new_game)
    game = new_game
```

```
void Person set_movie (string new_movie)
    movie = new_movie
```

```
void Person set_music (string new_music)
    music = new_music
```

```
void Person to_string ( ){
    Print first name, last name
    Print favorite movie genre
    Print favorite music genre
    Print favorite game genre
```

END Person.cpp file

START Friend.h file

```
include Person.h class
include iostream library
include array library
include namespace std
```

Start Friend class

Start public

```
Friend ( Build Array <Person, 5> & )
```

```
void update_friends ( )
```

```
void print_favorites ( )
```

```
void friend_of_friend ( )
```

End public

Start private

Build array <Person, 5> & friend_list

End private

End Friend class

END Friend.h file

START Friend.cpp file

```
include Friend.h class
include Person.h class
include iostream library
include array library
include namespace std
```

```
Friend Friend( array <Person, 5> & people )
    : friend_list (people)
```

```
void Friend update_friends ( )
    for (int x = 0; x < 5; x++) //iterated over 5 times
        if (friend_list[x].get_movie ( ) set as none)

            friend_list[x].set_movie(Person select_movie( ))
            friend_list[x].set_music(Person select_music( ))
            friend_list[x].set_game(Person select_game( ))
```

```
void Friend print_favs ( )
    for (int x = 0; x < 5; x++)
        Friend_list[x].to_string ( )
```

```

void Friend friend_of_friend ( )
    for (int x = 0; x < 4; x++)
        string favoritemovie = friend_list[x].get_movie ( )
        string favoritemusic = friend_list[x].get_music ( )
        string favoritegame = friend_list[x].get_game ( )

        for (int y = 1; y < 5; y++)
            string favoritemov= friend_list[y].get_movie ( )
            string favoritemus = friend_list[y].get_music ( )
            string favoritegam = friend_list[y].get_game ( )

            if ((favoritemovie == favoritemov) and (friend_list[x].get_first_name() !=
friend_list[y].get_first_name()))
                Print list of first names that share interest in movies

            if ((favorietmusic == favoritemus) && (friend_list[x].get_first_name() !=
friend_list[y].get_first_name()))
                Print list of first names that share an interest in music

            if ((favoritegame == favoritegam) && (friend_list[x].get_first_name() !=
friend_list[y].get_first_name()))
                Print list of first names that share an interest in games

```

END Friend.cpp file

END APPLICATION