

**Team 39 Term Project:**

Party Planner

Cindy Le, Courtney Meyer, James Moise, Dorothea Schmid, Sam Simon, Adam Tamargo

17 April 2017

*We, the members of Team 39, state that the code used in this project was written by us.*

COP 3503, Programming Fundamentals for CISE Majors II

Spring 2017

Prof. Jonathan C.L. Liu, Ph.D.

## **Abstract**

This report reflects the steps taken by our team at various levels of programming skill, planning, scheduling, and outlines the design and development of a computer software program written in C++ language. Our program “party planner” allows the user to plan an event and prints the resulting information into an invitation card using a template of the user’s choice along with an itinerary file containing information for the host. It is useful in order to help people plan events while efficiently utilizing their budgets and organizes the resulting information in a file output. Our specific party planning program is tested and compatible to run on the Windows 10 operating system. We also utilized multiple IDE’s for the development process, these include but are not limited to Visual Studio, Eclipse, CLion, and Code::Blocks. The design of the program makes use of abstract data types and software reuse. We also developed a file dedicated to bringing together all of the code we created; this combining file was the heart of our application, and thus was where our class specific code saw its usage. The problems and solutions to the problems taught us lessons for future works which are described below. Included in this final report is an introduction to the problem, an overview of the programming solution, testing techniques, possible extensions to the future development of this program, and a report conclusion. Following the report is an appendix which includes references used and a user manual.

## Table of Contents

Abstract.....	i
I.The Problem.....	1
II.Solution Overview.....	1
III.Program Testing.....	2
IV.Future Development.....	2
V.Conclusion.....	3
1. Lessons Learned.....	3
2. Future Work.....	3
VI.Appendices.....	4
1. References.....	4
2. User Manual.....	4

## **I. The Problem**

The specifications given were to form a team of seven people based on common interests in order to develop a C++ based software system in a team-based semester-long project. Guidelines for the project include a recommended length of three thousand lines of code for complexity. The program can be based on any computer machine or integrated development environment and must be presented to teaching assistants.

## **II. Solution Overview**

The vision was to create a software program that implemented topics and data structures learned in the past from COP3503 and COP3502. All possible project ideas included the use of file input and output, linked lists, vectors, and complex mathematic computations that were user friendly and practical. A party planning program was decided upon due to its ability to utilize a budget and organize information in a user-friendly way. The big concepts used were object orientation and inheritance for our classes. Some general techniques used were algorithmic development for the functions which required more computational analysis. Also, there is the usage of certain data structures which enabled our program to become more dynamic. In addition, we used a code splitting technique of utilizing many classes to create one big application.

### **III. Program Testing**

Static and dynamic approaches were used to test the program's efficiency. The most used technique was dynamic testing with test cases that are available in the appendices following this report. Some issues that appeared were errors dedicated to syntax. These errors ranged from incorrectly using certain methods to errors because of missing characters (commas, semicolons, etc.). There were also errors when implementing the code on a bigger scale. While each individual code segments may have compiled without errors, combining the segments resulted in errors for missing data types. Finally, we also ran into plenty of runtime errors while debugging and testing our application.

### **IV. Future Development**

Possibilities for extension of this program include the implementation of specifics regarding the guest's age, menu requests, location, etc. This can become specific down to each person. Questions such as if the person is 21 or older, the specific food they are eating, and how are they related to the host, could be answered to add a lot of extra useful features. Each specific type of event could be implemented as well. After all, a wedding party is a lot different than a birthday party. Beyond making the application more dynamic, there can be additional options for a user such as choosing an actual location of the party, directions to the party, and other useful options.

## **V. Conclusion**

### **1. Lessons Learned**

There are many aspects that come with the creation of a major project like this end of term project. Throughout our work towards our goal, our team found numerous struggles and successes.

First, the addition of multiple people working for the same end product adds complication when resolving errors due to differences in classes. However, we quickly found use in websites/programs such as GitHub which relieved us of such problems and aided in our communication. Along with the partially added confusion, we managed to fulfill the software development goal of each contributing to something we are proud of.

Besides teamwork, we have learned different techniques and methods to come to solutions for the problems that have existed. We utilized vectors, linked lists, and other unique styles that allowed us to complete such a task.

### **2. Future Work**

For future work, we have noticed the need for time management and the need for graphs/charts/plans such as using unified modeling languages. This project also taught us the valuable lesson of communication and teamwork which are essential to success in future projects, whether it be in class or in the professional workforce.

## **VI. Appendices**

### **1. References**

This report was written with reference to a PDF from Kyungpook National University on writing a software programming report ([http://ain.knu.ac.kr/RelatedLinks/data/Software\\_Programming\\_Project\\_and\\_Report\\_Writing.doc](http://ain.knu.ac.kr/RelatedLinks/data/Software_Programming_Project_and_Report_Writing.doc)). The template designs used for the final invitation card output came from FSymbols (<http://fsymbols.com>).

### **2. User Manual**

The program code was uploaded to and can be found on the University of Florida's CANVAS system. There are eight source files and six header files. The program begins by asking the user to input the name of the party, the budget, and the number of hours that the party will last. It then prompts the user to input the name of a text file containing first and last names of each attending guest. The main menu follows asking the user to determine either the location, entertainment, furniture, or food menu items for the party. Each choice has further specifications with each decision decreasing the remaining budget. The final output is an invitation card with details of the party and an itinerary file containing information for the host.