



2nd year 1st semester B.Sc. (Hons.) Final Examination- 2019

IT-2100: Semester Project and Viva

A Project report on

Class Routine Generator

A project report submitted to the Institute of Information Technology for partial fulfillment of the requirements for having the B.Sc.(Hons.) degree in Information Technology.

Submitted by:

Name	Exam Roll
Sumaiya Binte Shahid	180244
Maleeha Kaikaus	180249
A. K. Din Mohammad	180257
Moinul Islam Laskar	180270

Supervised by:

Manan Binte Taj Noor

Lecturer

Institute of Information Technology,

Jahangirnagar University, Savar, Dhaka-1342, Bangladesh.

DECLARATION

This project report is submitted to the Institute of Information Technology, Jahangirnagar University, Savar, Dhaka in partial fulfillment of the requirements for having the B.Sc.(Hons) degree in IT. This is also needed to certify that the project work is under the Second Year First Semester course “IT-2100: Semester Project and Viva”. So, we are here declaring that this project report has not been submitted elsewhere for the requirement of any kind of degree, diploma or publication.

Name	Exam Roll
Sumaiya Binte Shahid	180244
Maleeha Kaikaus	180249
A. K. Din Mohammad	180257
Moinul Islam Laskar	180270

ACCEPTANCE

This project report is submitted to the Institute of Information Technology, Savar, Dhaka in partial fulfillment of the requirements for having the B.Sc (Hons.) degree in Information Technology.

Manan Binth Taj Noor

Lecturer

Institute of Information Technology,
Jahangirnagar University, Savar, Dhaka-1342, Bangladesh.

ACKNOWLEDGEMENT

We would like to thank the Almighty for giving us the opportunity to complete this task. We express our sincere gratitude to all of them who provided insight and expertise that greatly assisted the research, although they may not agree with the interpretations/conclusions of this paper.

We thank, Manan Bintah Taj Noor, Lecturer of Institute of Information Technology, Jahangirnagar University, for supervising our project and relentlessly sharing her pearls of wisdom.

We would also like to show our gratitude to our honorable teachers who took time out to guide us and provide us with all the necessary materials and sufficient knowledge that was the major requirement.

Our acknowledgement is meant to express our sincere gratitude to all those people who have been associated with this project and have helped us with it by sharing their experiences and valuable opinions through which we received the required information crucial for our project.

ABSTRACT

Proper time management is the effective use of our time that allows us to plan our days in such a way that we finish our work with less effort and make the most of the limited time we've got. If there is a set activity such as class routine for every day, teacher will be able to focus more on their teaching and less on giving instructions and generally controlling the class. It will also help the students to manage their time and maintain punctuality. This is just two of many reasons, routines are important for teachers and students. Keeping these things in our mind, we have generated a class routine named "**Class Routine Generator**". This system is used to create and manipulate the class routine of an educational institution. By using it, our teachers will be able to create routines in less time and more efficiently and this will relieve them from doing manual activities. Sample routine of Institute of Information Technology of Jahangirnagar University was created by giving some courses with teachers' name and teachers' preferred time slot as input and we got a sample class routine as an output.

CONTENTS

Chapter 1: Introduction	07
1.1 Motivations	07
1.2 Objective	07
1.3 Contributions.....	07
 Chapter 2: Methodology.....	08
 Chapter 3: Brute Force Algorithm.....	10
 Chapter 4: Implementation.....	11
4.1 Algorithm	11
4.2 Flowchart	12
 Chapter 5: Results.....	13
 Chapter 6: Conclusion and Future Enhancement.....	21
6.1 Conclusions	21
6.2 Future Goal	21
 Chapter 7: References	22

Chapter 1

Introduction

1.1 Motivations

The main motivation of this project is to increase efficiency of a class routine and to reduce the manual activity of teachers. Time schedule is very important for our daily life. By making this program, we can develop our coding skill also. The knowledge obtained in this process can be applied later in the career to develop any similar kind of time management system.

1.2 Objectives

By developing this program, we will be able to –

- ☐ Build our own routine generator using **C++ language** [4][5], that we have learnt.
- ☐ Provide people a smarter and safer way to maintain their routine life.
- ☐ Be introduced with technological system like this organized class routine.

1.3 Contribution

To develop the time management system all the requirements are collected. Software, hardware and time management system requirement means also discussed in the chapter to initialize a cross platform software development library. After that move into our project to discuss about it in details.

Chapter 2

Methodology

For creating an efficient time management system, we studied about various method. [1] [3] By Brute force and many other methods, we have come to a solution for efficient and fast Class Routine Generator. To create the routine and to resolve collision between classes we have divided the scenario in different cases for a selected time. And they are:

Case 1: A class is free and teacher and the course related batch has no other class at that time.

Case 2: The Teacher has another class at that time.

Case 3: The course related batch has another class at that time.

Case 4: No classroom is free at that time.

For Case 1, the Teacher will be granted the class which he selects.

But Case 2, Case 3 and Case 4 come with 2 proposals.

Proposal 1: The teacher can select another time.

Proposal 2: The teacher can select the class by replacing another class. Such replacing can be done only two times.

Replacing of a class is different from cases to cases and they occur in following manner:

1. For case 2, the teacher will replace one of his two classes and that class will be asked for rescheduling later.
2. For case 3, the teacher who has been already scheduled for class of that particular batch will be replaced by the current Teacher. The replaced teacher will be asked for rescheduling later.

3. For case 4, the teacher who occupied the last class at that time will be replaced by the current teacher. The replaced teacher will be asked for rescheduling later.

When it is time for rescheduling, every teacher who has been replaced by other teacher will be asked again for reschedule their classes and their selected time will be again checked for Case 1 to 4. If collision resolve than they will be able to take class at that time.

Chapter 3

Brute Force Algorithms

Brute Force Algorithms refers to a programming style that does not include any shortcuts to improve performance, but instead relies on sheer computing power to try all possibilities until the solution to a problem is found.

A classic example is the traveling salesman problem (TSP). Suppose a salesman needs to visit 10 cities across the country. How does one determine the order in which cities should be visited such that the total distance traveled is minimized? The brute force solution is simply to calculate the total distance for every possible route and then select the shortest one. This is not particularly efficient because it is possible to eliminate many possible routes through clever algorithms.

Another example: 5-digit password, in the worst-case scenario would take 10^5 tries to crack.

The time complexity of brute force is $O(n*m)$. So, if we were to search for a string of 'n' characters in a string of 'm' characters using brute force, it would take us $n * m$ tries.

Implementing the brute-force search

Basic algorithm

In order to apply brute-force search to a specific class of problems, one must implement four procedures, *first*, *next*, *valid*, and *output*. These procedures should take as a parameter the data P for the particular instance of the problem that is to be solved, and should do the following:

1. *first* (P): generate a first candidate solution for P .
2. *next* (P, c): generate the next candidate for P after the current one c .
3. *valid* (P, c): check whether candidate c is a solution for P .
4. *output* (P, c): use the solution c of P as appropriate to the application.

The *next* procedure must also tell when there are no more candidates for the instance P , after the current one c . A convenient way to do that is to return a "null candidate", some conventional data value Λ that is distinct from any real candidate. Likewise, the *first* procedure should return Λ if there are no candidates at all for the instance P . [2]

Chapter 4

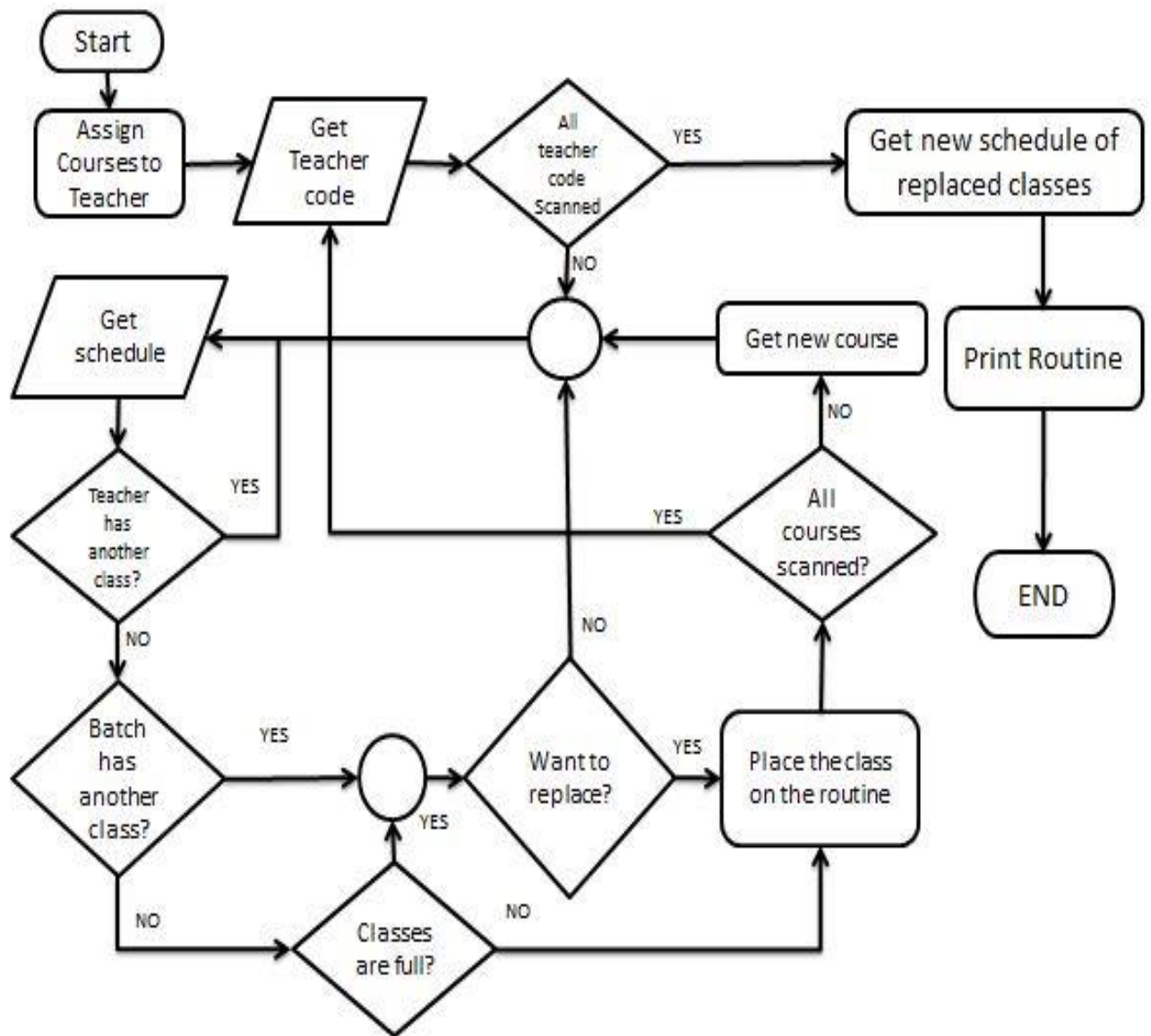
Implementation

4.1 Algorithm:

Algorithm for generating a class routine:

1. Assign Teachers to courses.
2. Scan Teacher Code
3. Repeat Step 4 to 6 for all courses which are assigned to teacher.
4. Get Class Schedule from teacher.
5. Check:
 - a) If the Teacher has another class at that time.
 - b) If the Batch has another class at that time.
 - c) If all the classes are full.
 - d) If any of a, b or c is true check, the teacher can replace any existing class or not.
6. If Step 5(d) is negative, go to step 4.
7. Else place the class schedule on the routine and replaced class to Reschedule Class List.
8. Go to step 2 if any teacher left to scan.
9. Get new schedule for every class in Reschedule Class List.
10. Print Class Routine.

4.2 Flowchart:

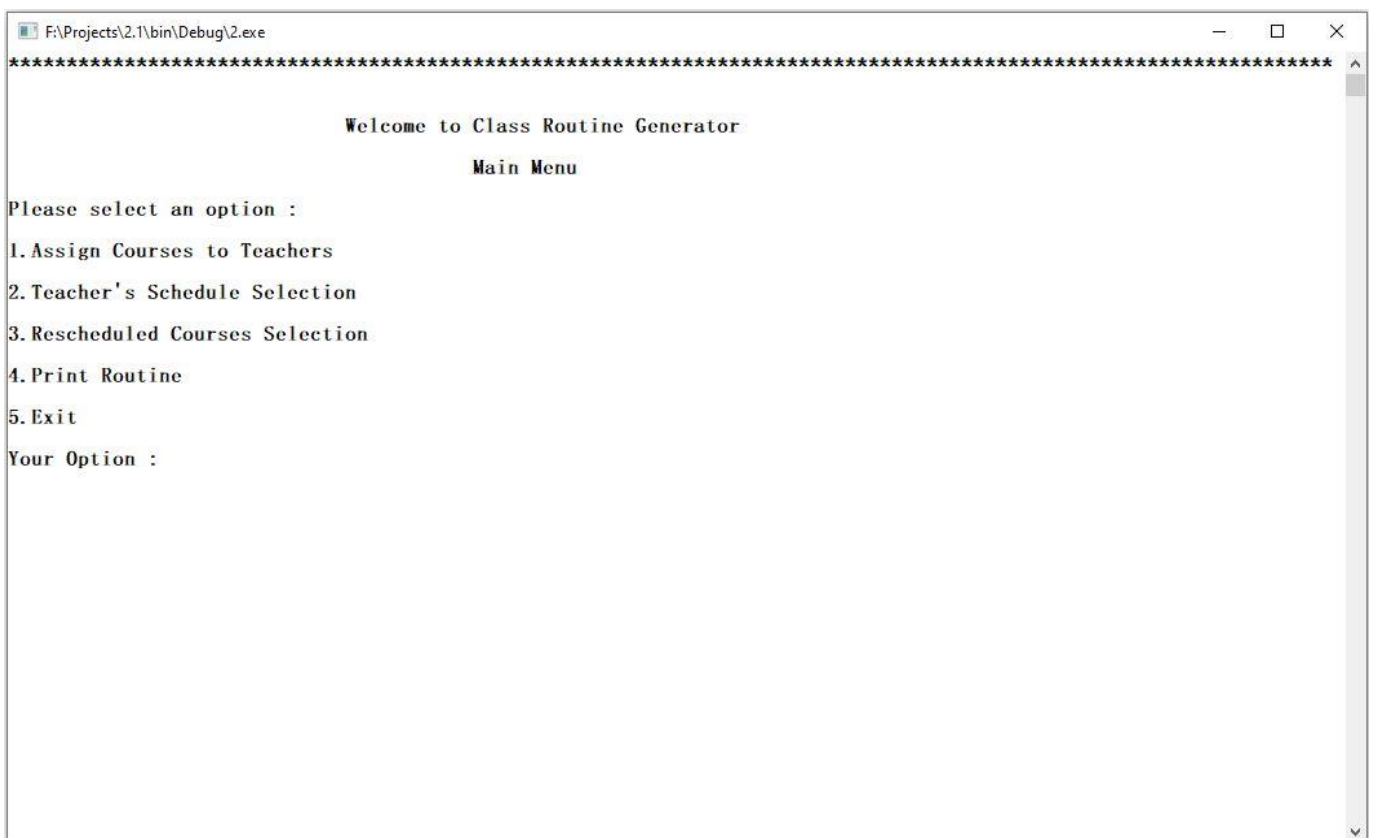


Chapter 5

Results

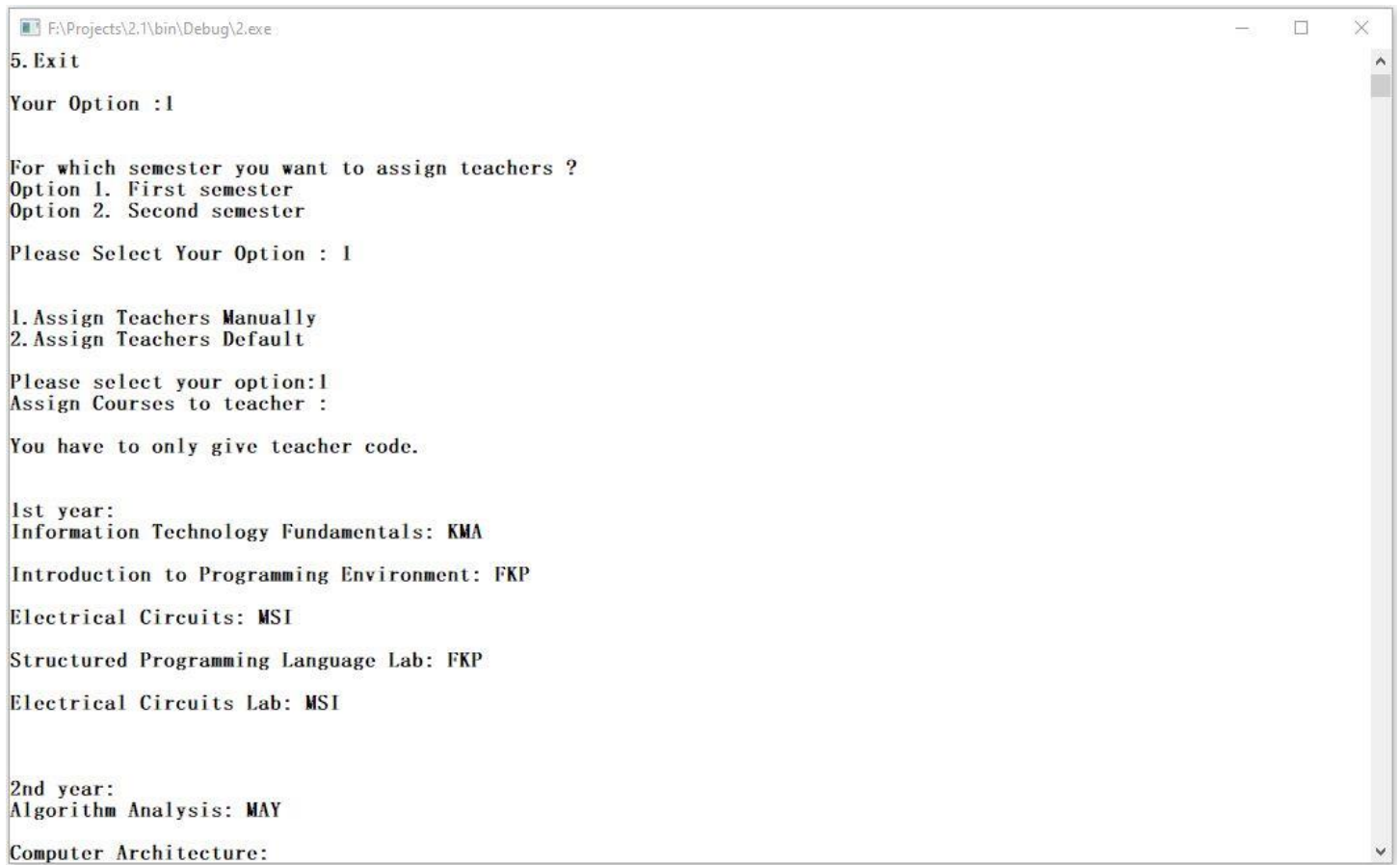
Figure of output of the project:

1. Routine Generator Starts.



```
F:\Projects\2.1\bin\Debug\2.exe
*****
Welcome to Class Routine Generator
Main Menu
Please select an option :
1.Assign Courses to Teachers
2.Teacher's Schedule Selection
3.Rescheduled Courses Selection
4.Print Routine
5.Exit
Your Option :
```

2. Assigning Teachers.



```
F:\Projects\2.1\bin\Debug\2.exe
5.Exit
Your Option :1

For which semester you want to assign teachers ?
Option 1. First semester
Option 2. Second semester

Please Select Your Option : 1

1.Assign Teachers Manually
2.Assign Teachers Default

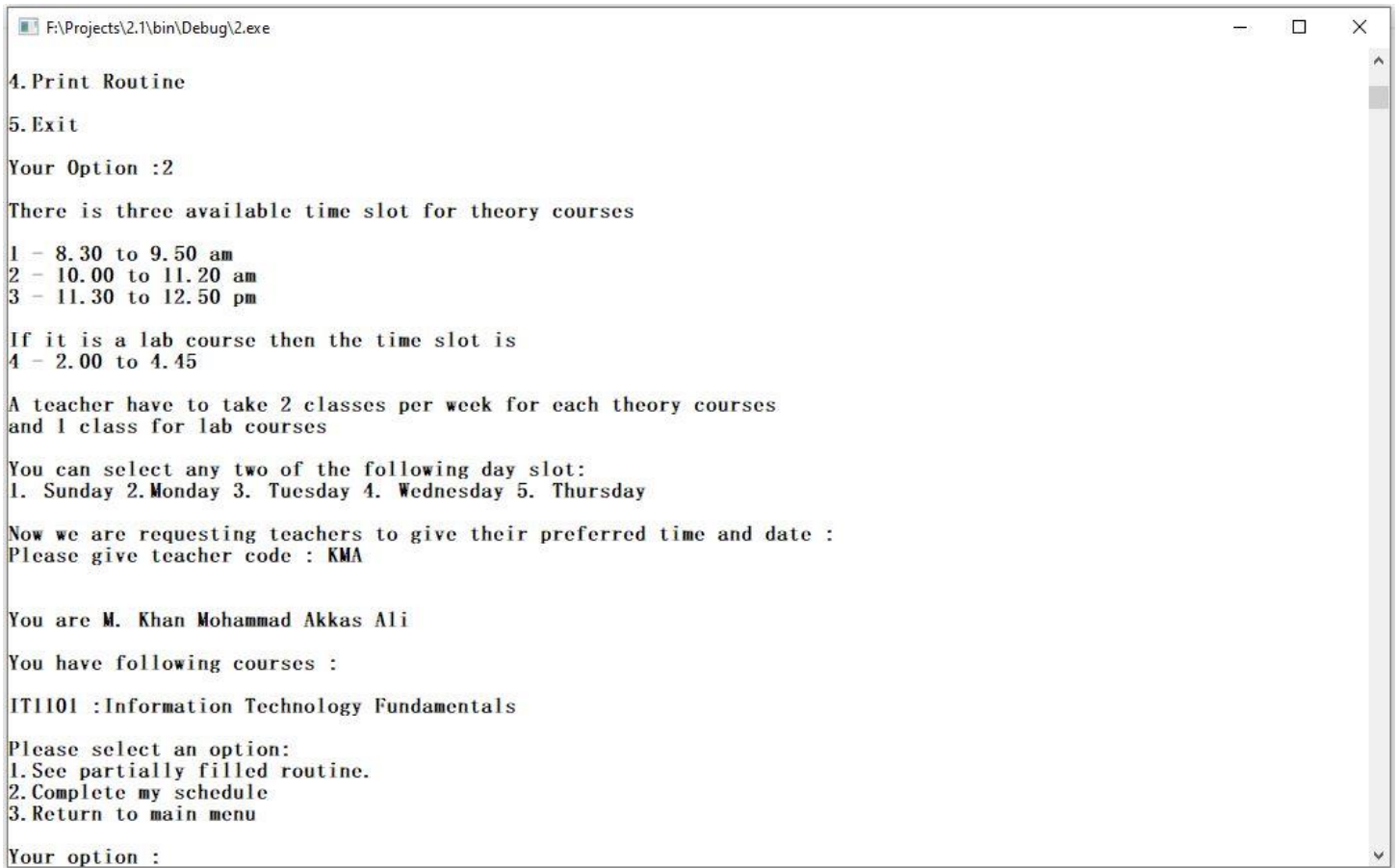
Please select your option:1
Assign Courses to teacher :

You have to only give teacher code.

1st year:
Information Technology Fundamentals: KMA
Introduction to Programming Environment: FKP
Electrical Circuits: MSI
Structured Programming Language Lab: FKP
Electrical Circuits Lab: MSI

2nd year:
Algorithm Analysis: MAY
Computer Architecture:
```

3. Showing Instructions with Teachers assigned courses.



```
F:\Projects\2.1\bin\Debug\2.exe

4. Print Routine
5. Exit
Your Option :2

There is three available time slot for theory courses
1 - 8.30 to 9.50 am
2 - 10.00 to 11.20 am
3 - 11.30 to 12.50 pm

If it is a lab course then the time slot is
4 - 2.00 to 4.45

A teacher have to take 2 classes per week for each theory courses
and 1 class for lab courses

You can select any two of the following day slot:
1. Sunday 2. Monday 3. Tuesday 4. Wednesday 5. Thursday

Now we are requesting teachers to give their preferred time and date :
Please give teacher code : KMA

You are M. Khan Mohammad Akkas Ali

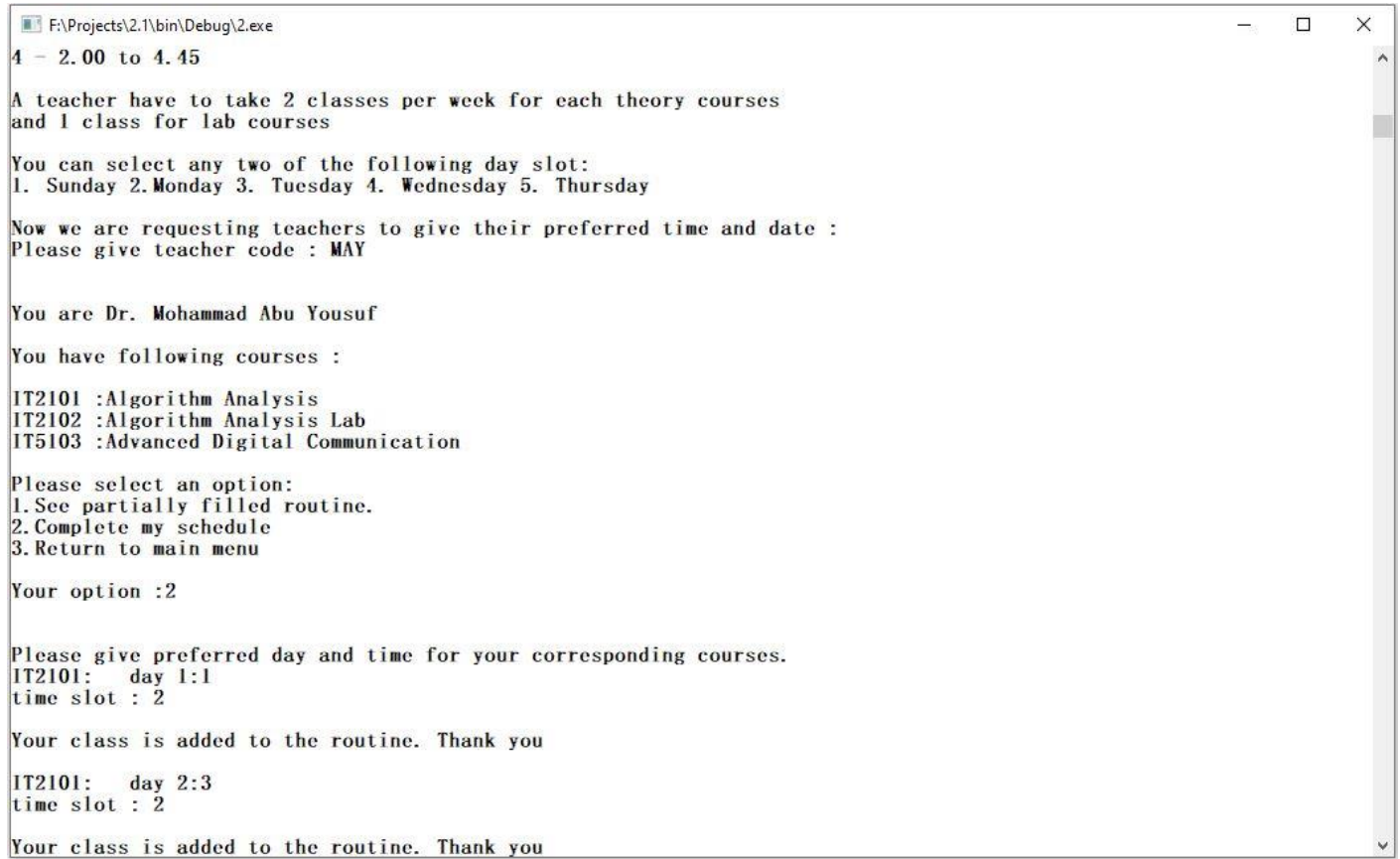
You have following courses :

IT1101 :Information Technology Fundamentals

Please select an option:
1. See partially filled routine.
2. Complete my schedule
3. Return to main menu

Your option :
```

4. Teachers Selecting Their Schedule



```
F:\Projects\2.1\bin\Debug\2.exe
4 - 2.00 to 4.45

A teacher have to take 2 classes per week for each theory courses
and 1 class for lab courses

You can select any two of the following day slot:
1. Sunday 2. Monday 3. Tuesday 4. Wednesday 5. Thursday

Now we are requesting teachers to give their preferred time and date :
Please give teacher code : MAY

You are Dr. Mohammad Abu Yousuf

You have following courses :

IT2101 :Algorithm Analysis
IT2102 :Algorithm Analysis Lab
IT5103 :Advanced Digital Communication

Please select an option:
1. See partially filled routine.
2. Complete my schedule
3. Return to main menu

Your option :2

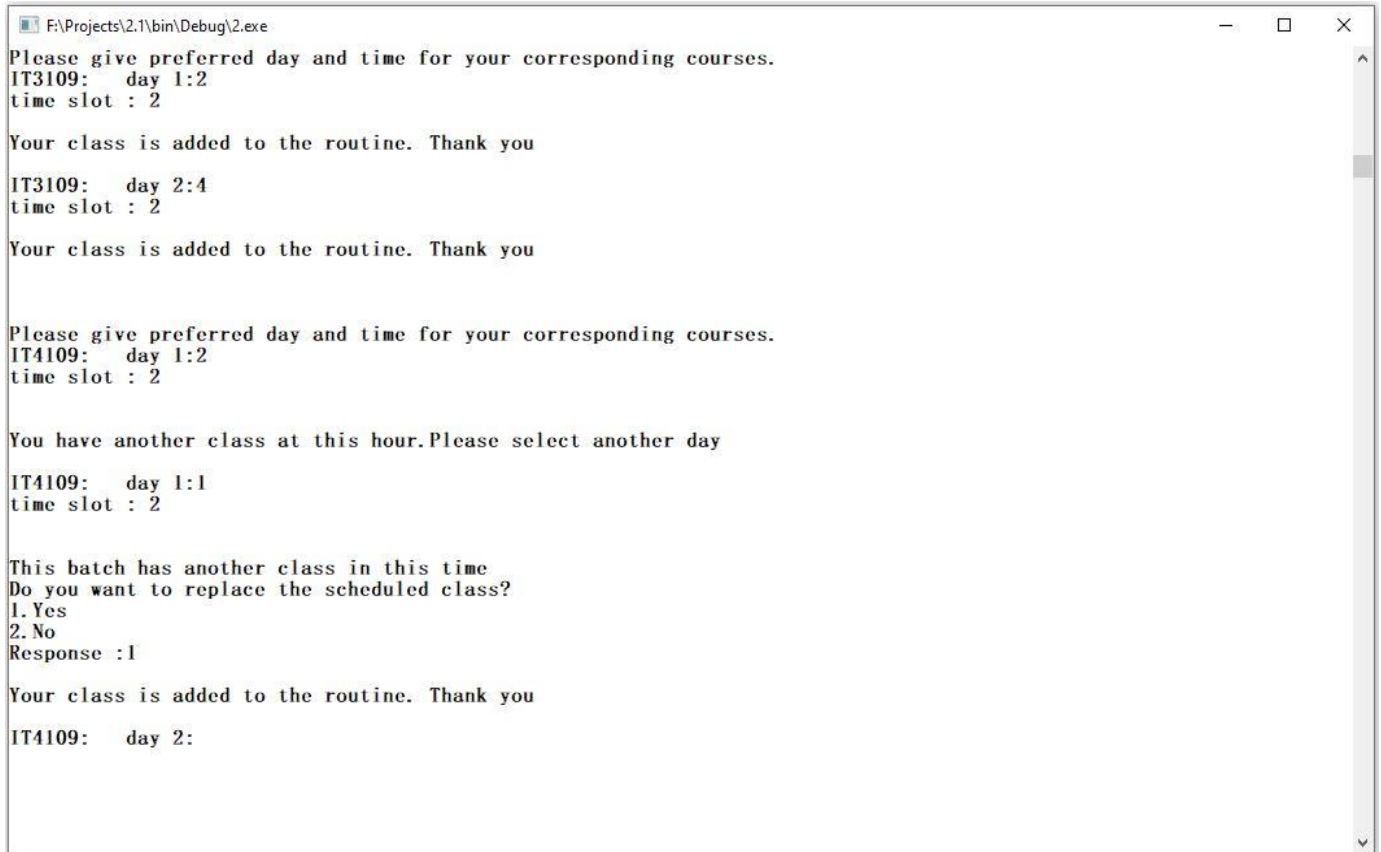
Please give preferred day and time for your corresponding courses.
IT2101: day 1:1
time slot : 2

Your class is added to the routine. Thank you

IT2101: day 2:3
time slot : 2

Your class is added to the routine. Thank you
```


5. A batch or Teacher is colliding with two classes at a time



```
F:\Projects\2.1\bin\Debug\2.exe
Please give preferred day and time for your corresponding courses.
IT3109:  day 1:2
time slot : 2

Your class is added to the routine. Thank you

IT3109:  day 2:4
time slot : 2

Your class is added to the routine. Thank you

Please give preferred day and time for your corresponding courses.
IT4109:  day 1:2
time slot : 2

You have another class at this hour.Please select another day

IT4109:  day 1:1
time slot : 2

This batch has another class in this time
Do you want to replace the scheduled class?
1.Yes
2.No
Response :1

Your class is added to the routine. Thank you

IT4109:  day 2:
```

6. Ongoing Routine

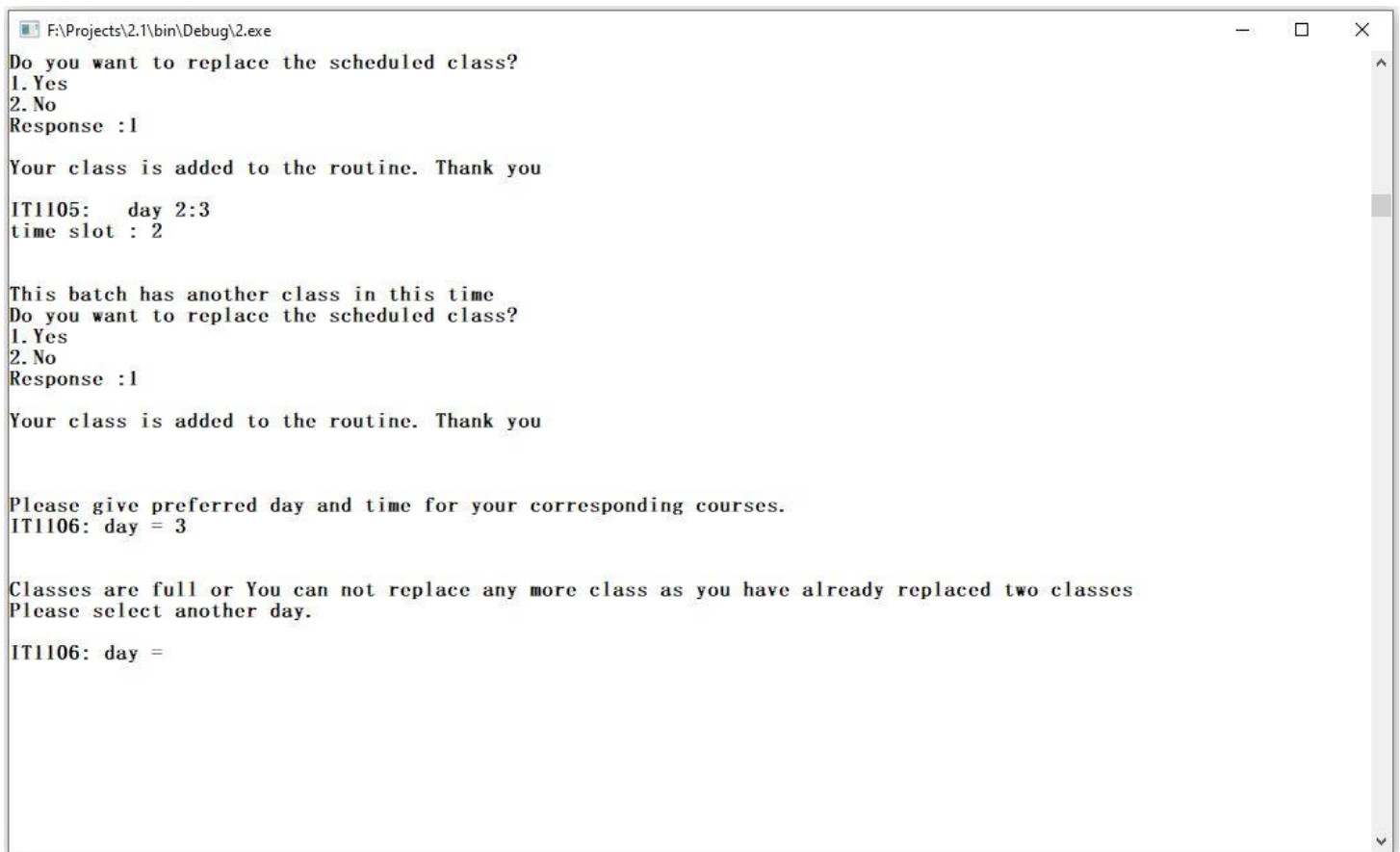
F:\Projects\2.1\bin\Debug\2.exe

Day	8.30-9.50	10.00-11.20	11.30-12.50	2.00-4.30
Sunday	IT1101(KMA), ()	IT1103(FKP), IT2101(MAY)	IT5103(MAY), ()	(), ()
	(), ()	IT4109(MSK), ()	(), ()	(), ()
Monday	IT1101(KMA), ()	IT1103(FKP), IT3109(MSK)	IT5103(MAY), IT4109(MSK)	IT1104(FKP), IT2102(MAY)
	(), ()	(), ()	(), ()	(), ()
Tuesday	(), ()	IT2101(MAY), IT4101(MMS)	(), ()	IT4102(MMS), ()
	(), ()	(), ()	(), ()	(), ()
Wednesday	(), ()	IT3109(MSK), ()	(), ()	(), ()
	(), ()	(), ()	(), ()	(), ()
Thursday	(), ()	(), ()	(), ()	(), ()
	(), ()	(), ()	(), ()	(), ()

Please select an option:
 1. See partially filled routine.
 2. Complete my schedule
 3. Return to main menu

Your option :

7. A Teacher can at most replace 2 classes



```
F:\Projects\2.1\bin\Debug\2.exe
Do you want to replace the scheduled class?
1. Yes
2. No
Response :1
Your class is added to the routine. Thank you
IT1105:  day 2:3
time slot : 2

This batch has another class in this time
Do you want to replace the scheduled class?
1. Yes
2. No
Response :1
Your class is added to the routine. Thank you

Please give preferred day and time for your corresponding courses.
IT1106: day = 3

Classes are full or You can not replace any more class as you have already replaced two classes
Please select another day.
IT1106: day =
```

8. Final Routine

Class Routine				
Day	8.30-9.50	10.00-11.20	11.30-12.50	2.00-4.30
Sunday	IT4103(RTK), IT3105(SAM)	IT2101(MAY), IT1101(KMA)	IT5109(WZ), IT1103(FKP)	IT1106(MSI), IT2102(MAY)
	(), ()	IT4101(MMS), ()	(), ()	IT3104(WZ), IT4104(RTK)
Monday	IT3103(WZ), ()	IT2105(SR), IT1105(MSI)	IT2103(RM), IT4105(MBTN)	IT2106(SR), ()
	(), ()	IT5101(MBTN), IT3107(FT)	IT5111(NZZ), IT3101(JA)	(), ()
Tuesday	IT3105(SAM), IT4109(MSK)	IT2101(MAY), IT1105(MSI)	IT5103(MAY), IT3109(MSK)	IT2104(RM), IT5102(SR)
	(), ()	IT4107(NZZ), IT3107(FT)	IT4101(MMS), ()	IT1104(FKP), IT4102(MMS)
Wednesday	IT4103(RTK), IT3103(WZ)	IT2103(RM), IT1103(FKP)	IT4105(MBTN), IT3101(JA)	IT3102(JA), ()
	(), ()	IT5101(MBTN), ()	IT5109(WZ), ()	(), ()
Thursday	IT4109(MSK), IT5103(MAY)	IT2105(SR), IT1101(KMA)	IT5111(NZZ), ()	IT3108(FT), ()
	(), ()	IT4107(NZZ), IT3109(MSK)	(), ()	(), ()

Chapter 6

Conclusion & Future Enhancement

6.1 Conclusion:

It is a complicated task to handle many faculties and allocate subjects for them at a time physically and the process is also error prone. So, our proposed system will help to overcome this disadvantage. Thus, we can produce a timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such a system we can make use of the different tools that are widely applicable and free to use also.

6.2 Future goal:

Our future endeavor will be:

- Code this program in Java with GUI (Graphical User Interface)
- Make this project more user friendly and efficient.
- Enhance this project to work on any type of data set.

Chapter 7

Reference

- [1] Wikipedia contributors. (2019, September 9). Time management. In Wikipedia, Available: https://en.wikipedia.org/w/index.php?title=Time_management&oldid=914775407. [Accessed: 13- Sep- 2019].
- [2] Wikipedia contributors, "Brute-force attack," Available: https://en.wikipedia.org/w/index.php?title=Bruteforce_attack&oldid=914861211 [Accessed September 12, 2019]
- [3] Wikipedia contributors, "Scheduling (computing)," Available: [https://en.wikipedia.org/w/index.php?title=Scheduling_\(computing\)&oldid=913553290](https://en.wikipedia.org/w/index.php?title=Scheduling_(computing)&oldid=913553290) [Accessed: September 14, 2019]
- [4] E Balagurusamy, Balaguruswamy OOP with C++, 2008
- [5] John R. Hubbard, McGraw-Hill Education, Schaum's Outline of Programming With C++, 2nd edition (June 6, 2000)