Approach for Automatic Timetable is as follow:

The program read information's from file to classes we first stored each line. Then we split each line to string by ";" since in our file it's been separated by ";". Priority is to service courses in the table. So when generating table the program looks first for service courses when program get service courses then it check for available day from table, check for available time from table, check for elective or mandatory, if it elective the program assigns it to big or small class in table otherwise assign course to big class in table.

Next, the program takes department course class and check for busy course (busy time), if the time slot is busy the program look for available day, then available time and check for elective or mandatory, if the subject is elective the program assign course to free big or small class in table otherwise if its mandatory the program assign course to free big class in table. The program goes so on for all subjects and remaining classes. If program can't make table, it prints message to screen "There is no way to make a perfect schedule for the department."

a. Exception handling: In this program, exception handling has been used for a function to check if the index for strings cross boundaries. If it passes, then the program throw exception and print a message on screen ("index of split string array is out of boundary"). The code shows more accurate.

```
void Store_Course_Info_To_Vector(QVector<course>& new_Course, int columns =
0,int rows = 0 ){
  read_Myfile read_courses_csv_file;
          QString file_Name="C:/Users/DELL/Desktop/PL 2nd
checkpoint/table/courses.csv";
  read_courses_csv_file.split_To_String(file_Name);
          try {
    while(rows < read_courses_csv_file.split_To_String(file_Name).size()){
              course
newCourse(read_courses_csv_file.split_To_String(file_Name)[rows][columns],
read_courses_csv_file.split_To_String(file_Name)[rows][columns+1],

read_courses_csv_file.split_To_String(file_Name)[rows][columns+2],
read_courses_csv_file.split_To_String(file_Name)[rows][columns+3],

read_courses_csv_file.split_To_String(file_Name)[rows][columns+4],
read_courses_csv_file.split_To_String(file_Name)[rows][columns+5],

read_courses_csv_file.split_To_String(file_Name)[rows][columns+6]);
                new_Course.push_back(newCourse);
        rows++;
    }if(rows > read_courses_csv_file.split_To_String(file_Name).size()){
        throw "read_courses_csv_file out of boundary";
           }
  } catch (char *ex) {
  cout<< ex;
 }
}
```

b. Default Parameter: The program has (bool **is_CourseMorning_Busy)** function as default parameter. The code shows it more proper…

```
bool is_CourseMorning_Busy(course& value_of_Course, int index_Of_Day,
QVector<QVector<QString>>& time_Slot_Busy, bool value_To_Be_Returned =
false)
  {

 int j = 0;
 while(j < time_Slot_Busy.size()){
      int index_Busy_TimeSlot = get_Number_Of_Day(time_Slot_Busy[j][1]);
```

```
        if ( index_Of_Day == index_Busy_TimeSlot &&
  value_of_Course.get_Course_Code() == time_Slot_Busy[j][0] &&
             time_Slot_Busy[j][2] == "Morning" )
      {
          value_To_Be_Returned = true;
          break;
      }
      j++;
  }
    return value_To_Be_Returned;
}
```

c. Operating Overloading: The program has Operating overloading feature in course class. The code is listed below…

```
course();

course(QString,  QString,  QString,  QString,QString,  QString,
QString);
```

d. Inline function: This feature has been used for (**get_Number_Of_Day**) function.

```
inline int get_Number_Of_Day(QString dayValue){
 if(dayValue=="Monday") return 0;
if(dayValue=="Tuesday") return 1;
 if(dayValue=="Wednesday") return 2;
 if(dayValue=="Thursday") return 3;
 if(dayValue=="Friday") return 4;
else return -1;
}
```

e. Constructor/ Destructor:

```
//default constructor
 course();
//~course(); destructor
 ~course();
```

f. Vectors: The program has vector for reading course.csv file…

```
void Store_Course_Info_To_Vector(QVector<course>& new_Course, int columns =
0,int rows = 0 ){
  read_Myfile read_courses_csv_file;
          QString file_Name="C:/Users/DELL/Desktop/PL 2nd
checkpoint/table/courses.csv";
  read_courses_csv_file.split_To_String(file_Name);
          try {
   while(rows < read_courses_csv_file.split_To_String(file_Name).size()){
```

```
                course
        newCourse(read_courses_csv_file.split_To_String(file_Name)[rows][columns],
        read_courses_csv_file.split_To_String(file_Name)[rows][columns+1],

        read_courses_csv_file.split_To_String(file_Name)[rows][columns+2],
        read_courses_csv_file.split_To_String(file_Name)[rows][columns+3],

        read_courses_csv_file.split_To_String(file_Name)[rows][columns+4],
        read_courses_csv_file.split_To_String(file_Name)[rows][columns+5],

        read_courses_csv_file.split_To_String(file_Name)[rows][columns+6]);
                    new_Course.push_back(newCourse);
                rows++;
           }if(rows > read_courses_csv_file.split_To_String(file_Name).size()){
                throw "read_courses_csv_file out of boundary";
                   }
          } catch (char *ex) {
          cout<< ex;
         }
        }
```

Our classes files are as follow:

Classroom.cpp
Classroom.h

Day.h
Day.cpp

Main.cpp

Professor.cpp
Professor.h

Readmyfile.cpp
Readmyfile.h

Timeslot.cpp
Timeslot.h

Writeresulttofile.cpp
Writeresulttofile.h

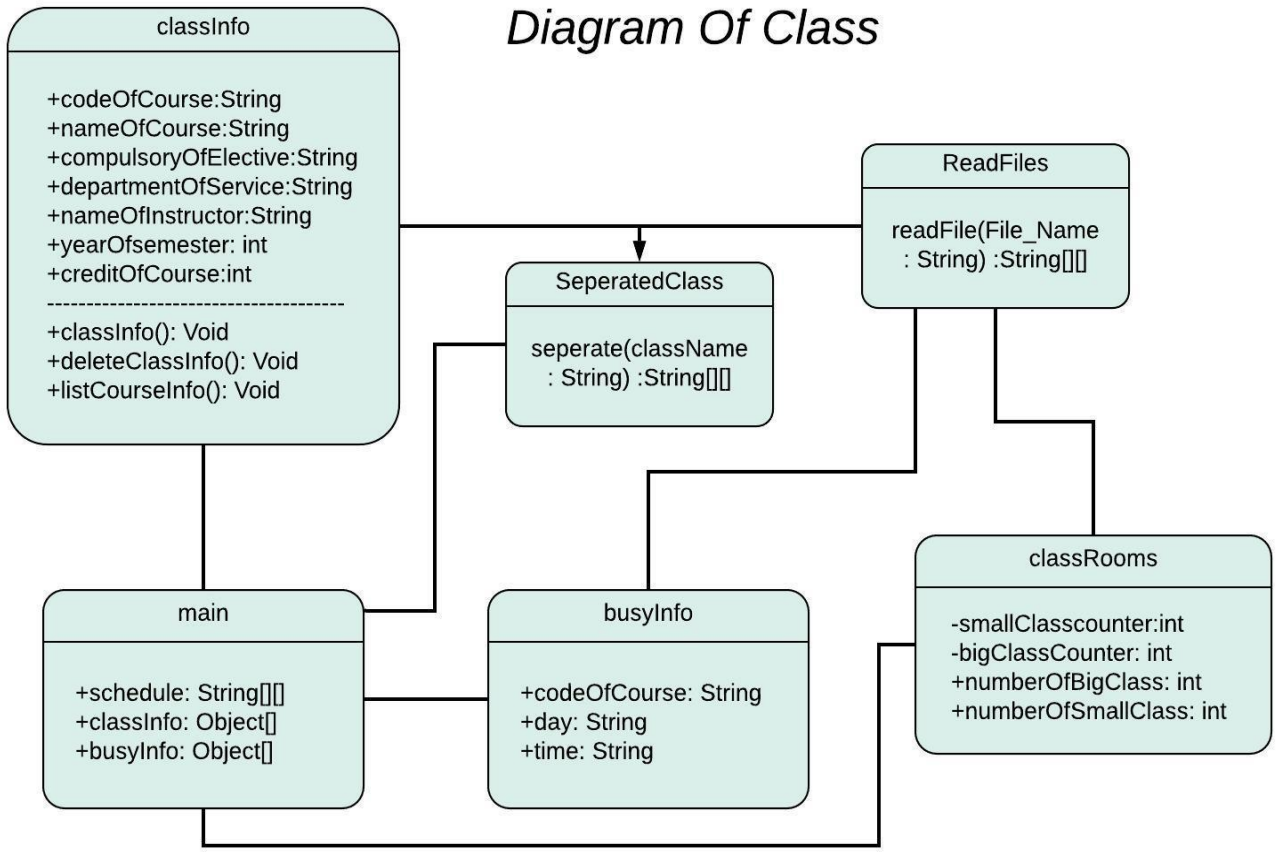
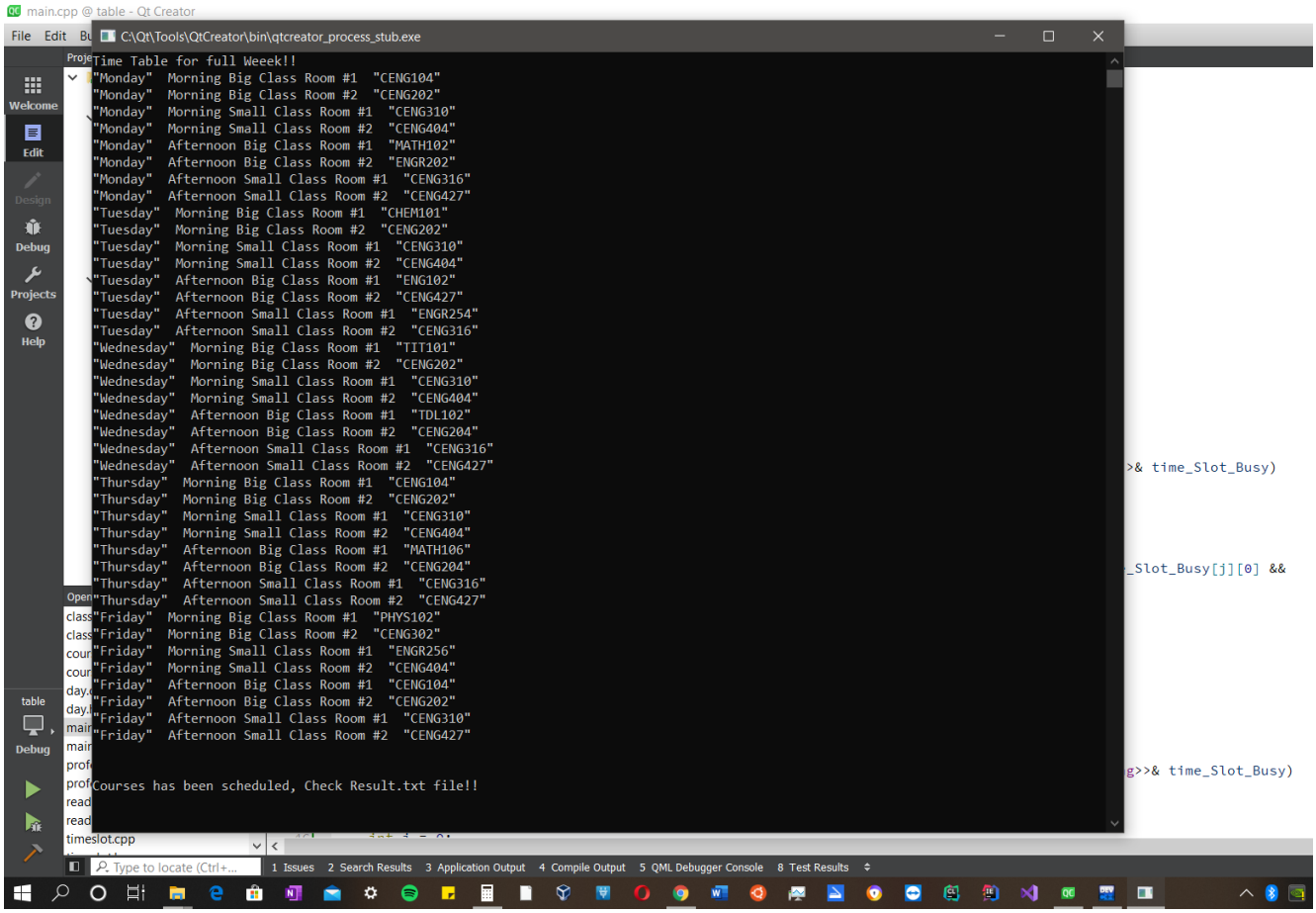


Diagram Of Class
classInfo
+codeOfCourse:String
+nameOfCourse:String
+compulsoryOfElective:String
+departmentOfService:String
+nameOfInstructor:String
+yearOfsemester: int
+creditOfCourse:int
+classInfo(): Void
+deleteClassInfo(): Void
+listCourseInfo(): Void
ReadFiles
readFile(File_Name : String) :String[][]
SeperatedClass
seperate(className : String) :String[][]
classRooms
-smallClasscounter:int
-bigClassCounter: int
+numberOfBigClass: int
+numberOfSmallClass: int
main
+schedule: String[][]
+classInfo: Object[]
+busyInfo: Object[]
busyInfo
+codeOfCourse: String
+day: String
+time: String

Output...


main.cpp @ table - Qt Creator
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Time Table for full Weeek!!
"Monday" Morning Big Class Room #1 "CENG104"
"Monday" Morning Big Class Room #2 "CENG202"
"Monday" Morning Small Class Room #1 "CENG310"
"Monday" Morning Small Class Room #2 "CENG404"
"Monday" Afternoon Big Class Room #1 "MATH102"
"Monday" Afternoon Big Class Room #2 "ENGR202"
"Monday" Afternoon Small Class Room #1 "CENG316"
"Monday" Afternoon Small Class Room #2 "CENG427"
"Tuesday" Morning Big Class Room #1 "CHEM101"
"Tuesday" Morning Big Class Room #2 "CENG202"
"Tuesday" Morning Small Class Room #1 "CENG310"
"Tuesday" Morning Small Class Room #2 "CENG404"
"Tuesday" Afternoon Big Class Room #1 "ENG102"
"Tuesday" Afternoon Big Class Room #2 "CENG427"
"Tuesday" Afternoon Small Class Room #1 "ENGR254"
"Tuesday" Afternoon Small Class Room #2 "CENG316"
"Wednesday" Morning Big Class Room #1 "TIT101"
"Wednesday" Morning Big Class Room #2 "CENG202"
"Wednesday" Morning Small Class Room #1 "CENG310"
"Wednesday" Morning Small Class Room #2 "CENG404"
"Wednesday" Afternoon Big Class Room #1 "TDL102"
"Wednesday" Afternoon Big Class Room #2 "CENG204"
"Wednesday" Afternoon Small Class Room #1 "CENG316"
"Wednesday" Afternoon Small Class Room #2 "CENG427"
"Thursday" Morning Big Class Room #1 "CENG104"
"Thursday" Morning Big Class Room #2 "CENG202"
"Thursday" Morning Small Class Room #1 "CENG310"
"Thursday" Morning Small Class Room #2 "CENG404"
"Thursday" Afternoon Big Class Room #1 "MATH106"
"Thursday" Afternoon Big Class Room #2 "CENG204"
"Thursday" Afternoon Small Class Room #1 "CENG316"
"Thursday" Afternoon Small Class Room #2 "CENG427"
"Friday" Morning Big Class Room #1 "PHYS102"
"Friday" Morning Big Class Room #2 "CENG302"
"Friday" Morning Small Class Room #1 "ENGR256"
"Friday" Morning Small Class Room #2 "CENG404"
"Friday" Afternoon Big Class Room #1 "CENG104"
"Friday" Afternoon Big Class Room #2 "CENG202"
"Friday" Afternoon Small Class Room #1 "CENG310"
"Friday" Afternoon Small Class Room #2 "CENG427"
Courses has been scheduled, Check Result.txt file!!
timeslot.cpp
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 QML Debugger Console
8 Test Results