```
using System;
using System.Collections.Generic;
using System.Text;
using System.Collections;
using System.IO;

namespace StudentsManagement
{
   class Analyzer
   {
      Classroom classroom;
      const string path = @"D:\c#projects\StudentsManagement\data\exports\";
      public Analyzer(Classroom classroom)
      {
         this.classroom = classroom;
      }
      public void printStudentsData()
      {
         classroom.printStudents();
      }
      public void printBetterStudents()//גבוה או שווה לקודמי.  מדפיסה תלמידים שציונם במבחן האחרון\
      {
         Console.ForegroundColor = ConsoleColor.Green;
         Console.WriteLine("Better Students: ");
         Console.ForegroundColor = ConsoleColor.White;
         Console.WriteLine("---------------- \n");
         int cnt=0;
         ArrayList students = classroom.getStudents();
         foreach(student student in students)
         {
            grade[] gradesList = student.getAllGrades();

            foreach(grade grade in gradesList)
            {
               ArrayList grades = grade.getGrades();
               int high = 0;
               foreach(int speGrade in grades)
               {
```

```
            if (high < speGrade)
                high = speGrade;
        }
        object[] tempAl = grades.ToArray();
        int tempLastGrade = int.Parse((tempAl[grade.getNumOfGrades() - 1]).ToString());
        if (tempLastGrade == high)
        {
            Console.WriteLine("----------------");
            student.printID();
            Console.WriteLine(classroom.getSubjectNameByID(grade.getSubID())+
                " :"+ high);
            Console.WriteLine("----------------");
            cnt = cnt+1;
        }
      }
   }
   if (cnt==0)
      Console.WriteLine("Not Found! :(");

   Console.WriteLine("\n Export? (yes/no)");
   string answer = Console.ReadLine();
   if(answer == "yes")
   {
      exportPrintBetterStudents();
   }
   Console.WriteLine(" \n ---------------------------------- \n ");



}
 private void exportPrintBetterStudents()
{
   int index = 0;
   string tempPath = path + "PrintStudents" + index.ToString();
   while (File.Exists(tempPath))
   {
      index = index + 1;
      tempPath = path + "PrintStudents" + index;
   }
```

```
    using (StreamWriter sw = new StreamWriter(tempPath))
    {
        ArrayList students = classroom.getStudents();
        foreach (student student in students)
        {
            grade[] gradesList = student.getAllGrades();

            foreach (grade grade in gradesList)
            {
                ArrayList grades = grade.getGrades();
                int high = 0;
                foreach (int speGrade in grades)
                {
                    if (high < speGrade)
                        high = speGrade;
                }
                object[] tempAl = grades.ToArray();
                int tempLastGrade = int.Parse((tempAl[grade.getNumOfGrades() - 1]).ToString());
                if (tempLastGrade == high)
                {
                    //student.printID();
                    sw.WriteLine(student.getID());
                    sw.WriteLine(classroom.getSubjectNameByID(grade.getSubID()) +
                        " :" + high);
                    sw.WriteLine(" ");
                }
            }
        }
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine(" \n Exported!! :)");
        Console.ForegroundColor = ConsoleColor.White;
        Console.WriteLine("\n check out: "+tempPath);
    }
}
public void printStudentsAbove()
{
    Console.WriteLine("\n");
```

```
Console.ForegroundColor = ConsoleColor.Green;
Console.WriteLine("Students with higher grade: ");
Console.ForegroundColor = ConsoleColor.White;
Console.WriteLine("---------------------------- \n");
int minGrade;
int subjectID;
int cnt = 0;
Console.WriteLine("Enter minimal grade:\n");
minGrade = int.Parse(Console.ReadLine());
Console.WriteLine("[1- Math 2- English 3- Physics]");
Console.WriteLine("Enter Subject ID:\n");
subjectID = int.Parse(Console.ReadLine());
Console.WriteLine(" ----------");
ArrayList students = classroom.getStudents();
foreach (student student in students)
{
   grade[] gradesList = student.getAllGrades();
   ArrayList grades = gradesList[subjectID - 1].getGrades();
   foreach (int speGrade in grades)
   {
      if (minGrade <= speGrade)
      {
         student.printID();
         Console.WriteLine("his grade: " + speGrade+"\n ----------");

         cnt = cnt + 1;
      }
   }
}
if (cnt == 0)
   Console.WriteLine("Not Found! :(");

Console.WriteLine("\n Export? (yes/no)");
string answer = Console.ReadLine();
if (answer == "yes")
{
   exportPrintStudentsAbove(minGrade,subjectID);
}
```

```
            Console.WriteLine(" \n ---------------------------------- \n ");
        }
        private void exportPrintStudentsAbove(int minGrade, int subjectID)
        {
            int index = 0;
            string tempPath = path + "StudentsAbove" + index.ToString();
            while (File.Exists(tempPath))
            {
                index = index + 1;
                tempPath = path + "StudentsAbove" + index;
            }
            using (StreamWriter sw = new StreamWriter(tempPath))
            {
                sw.WriteLine("minimum grade: " + minGrade + " in subject: " + classroom.getSubjectNameByID(subjectID -
1));
                sw.WriteLine("");
                ArrayList students = classroom.getStudents();
                foreach(student student in students)
                {
                    ArrayList grades = student.getGradesByID(subjectID).getGrades();
                    foreach(int specGrade in grades)
                    {
                        if (specGrade>= minGrade)
                        {
                            sw.WriteLine(student.getID());
                            sw.WriteLine("his grade was: " + specGrade+"\n");
                        }
                    }
                }
            }
            Console.ForegroundColor = ConsoleColor.Green;
            Console.WriteLine(" \n Exported!! :)");
            Console.ForegroundColor = ConsoleColor.White;
            Console.WriteLine("\n check out: " + tempPath);
        }
    }
}
```