ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 1: Programming		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Bùi Hương Linh	**Student ID**	GBH200662
Class	GCH1002	**Assessor name**	Lecturer Mạnh

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	

Grading grid

P1	M1	D1

☼Summative Feedback:	☼Resubmission Feedback:

Grade:	Assessor Signature:	Date:

Lecturer Signature:

Contents

Task 1: Provide a definyion of what an algorithm is and represent a small and simple problem.

I. Algorithm

I.1/ What is an algorithm ?

- Algorithm is a set of instructions on how to carry out a process that is clearly defined instructions of the sequence of activities that comprise a process of obtaining the desired outputs from the provided inputs.(Riya Kumari, 2021)

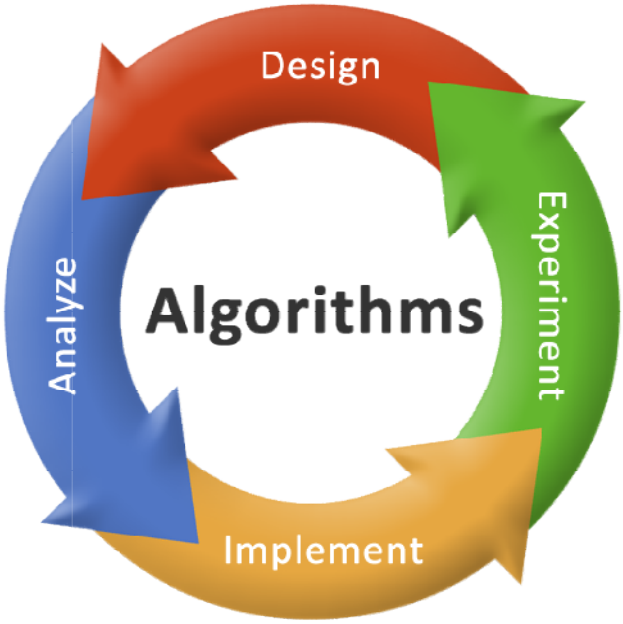


Image 1: Algorithms (source: Internet)

I.2/ Characteristics of Algorithm (Riya Kumari, 2021)

- Input: The input is the information that will be altered throughout the computation to produce the outcome.
- Output: The output is the information that results from the calculation.
- Clear and Unambiguous: Algorithms must determine each step and each step must be clear in all behaviors and point to just one meaning.
- Feasible: The algorithm should be effective which implies that all those means that are needed to get to output must be feasible with the accessible resources.
- Independent:n An algorithm should have step-by-step instructions that are not dependent on any programming code.
- Finiteness: An algorithm should have step-by-step instructions that are not dependent on any programming code.

I.3/ Advantages of Algorithm (Copyright Way2Benefits, 2021)

- It is not dependant on any programming language, making it simple to grasp for everyone, including those with no programming experience.
- An algorithm follows a certain method.
- It is simple to create an algorithm, then translate it into a flowchart, and finally into a computer program.
- An algorithm serves as a blueprint for a program and aids in its development.
- Each step in an algorithm has its own logical sequence, making it simple to debug.
- Easy to understand.

I.4/ Disad vantages of Algorithm (Copyright Way2Benefits)

- Time-consuming.

- Understanding complex logic via algorithms can be challenging.
- Algorithms for large jobs are difficult to implement.
- It is difficult to show branching and looping in algorithms.

II. Small and simple problem : Insertion Sort

- What is insertion sort ?
  - Insertion Sort is a sorting algorithm that takes an element at a time and inserts it in its correct position in the array. This procedure is repeated until the array is sorted. (Arjun Thakur, 2021)

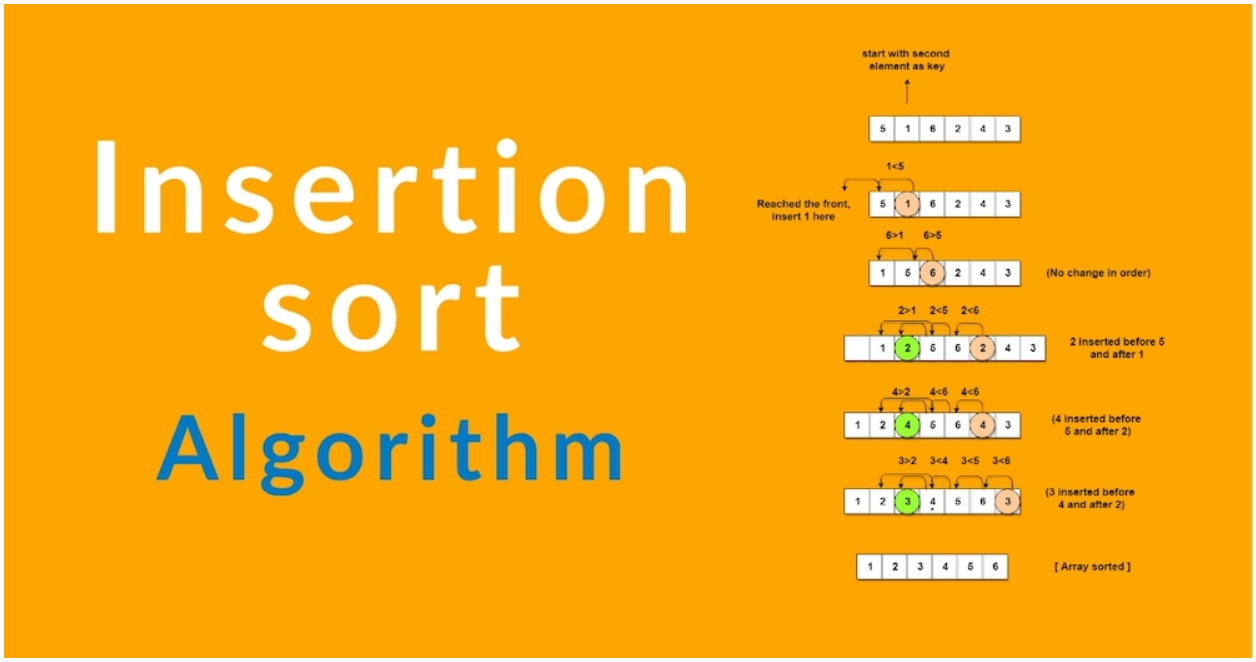
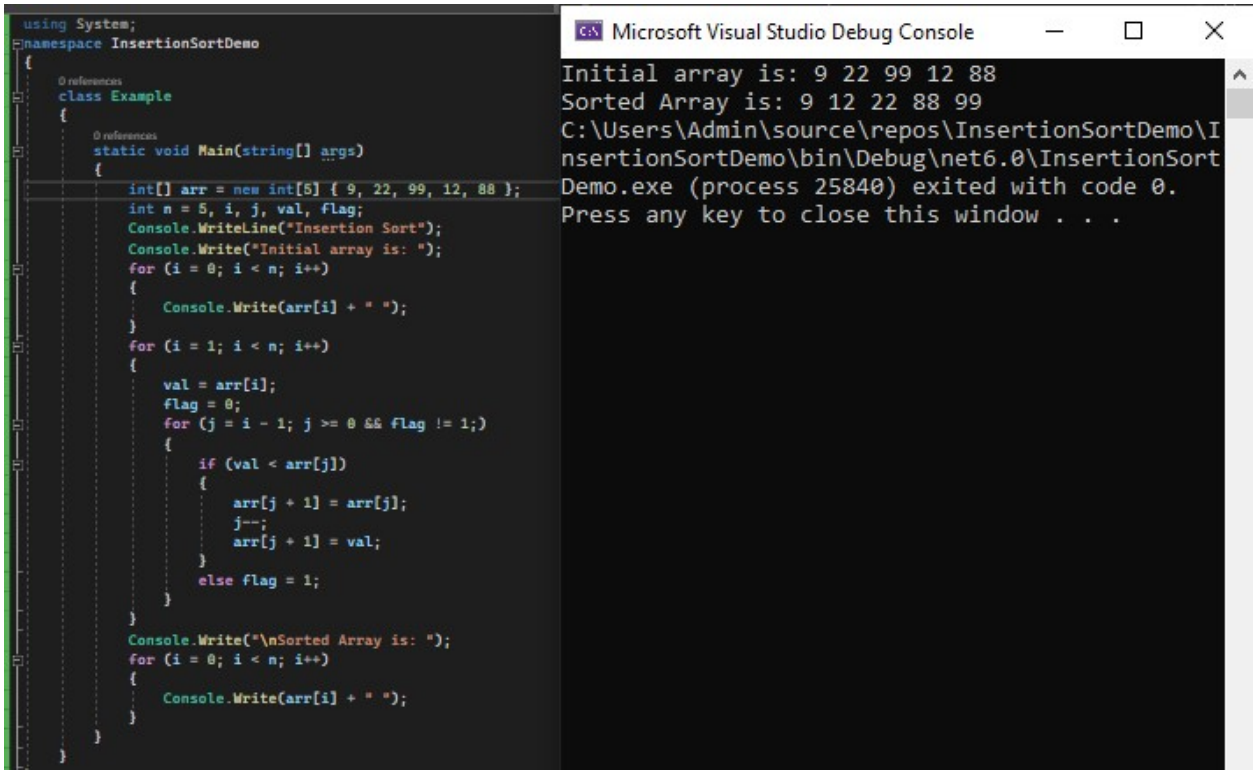


Image 2: Insertion sort (source: Internet)

- Example:
  Input: Initial array is: 9 22 99 12 88
  Output: Sorted array is: 9 12 22 88 99



```
using System;
namespace InsertionSortDemo
{
    0 references
    class Example
    {
        0 references
        static void Main(string[] args)
        {
            int[] arr = new int[5] { 9, 22, 99, 12, 88 };
            int n = 5, i, j, val, flag;
            Console.WriteLine("Insertion Sort");
            Console.Write("Initial array is: ");
            for (i = 0; i < n; i++)
            {
                Console.Write(arr[i] + " ");
            }
            for (i = 1; i < n; i++)
            {
                val = arr[i];
                flag = 0;
                for (j = i - 1; j >= 0 && flag != 1;)
                {
                    if (val < arr[j])
                    {
                        arr[j + 1] = arr[j];
                        j--;
                        arr[j + 1] = val;
                    }
                    else flag = 1;
                }
            }
            Console.Write("\nSorted Array is: ");
            for (i = 0; i < n; i++)
            {
                Console.Write(arr[i] + " ");
            }
        }
    }
```

```
Microsoft Visual Studio Debug Console
Initial array is: 9 22 99 12 88
Sorted Array is: 9 12 22 88 99
C:\Users\Admin\source\repos\InsertionSortDemo\I
nsertionSortDemo\bin\Debug\net6.0\InsertionSort
Demo.exe (process 25840) exited with code 0.
Press any key to close this window . . .
```

We see very clearly that the elements of the array after running the program have been sorted in ascending order, for better understanding I will analyze each step: (Arjun Thakur, 2021)

- Step 1: First the array is initialized and its value is printed using a for loop.

- Step 2: A nested for loop is used for the actual sorting process. In each pass of the outer for loop, the current element is inserted into its correct position in the array.

- Step 3: Finally, the sorted array is displayed.

Task 2: Analyse the problem and design the solutions by the use of suitable methods.

I. Problem

- Create a student management software program:The software includes student information, 3 subject scores, academic performance, average, and menus. The menu contains 6 cases.

II. Solution

- To solve the problem, I wrote a student management application in C#. I use classes and objects in my program because a class is a pattern or design, and an object is created. In this program, class will contain an ID, name, age,gender,...

III. Flowchart

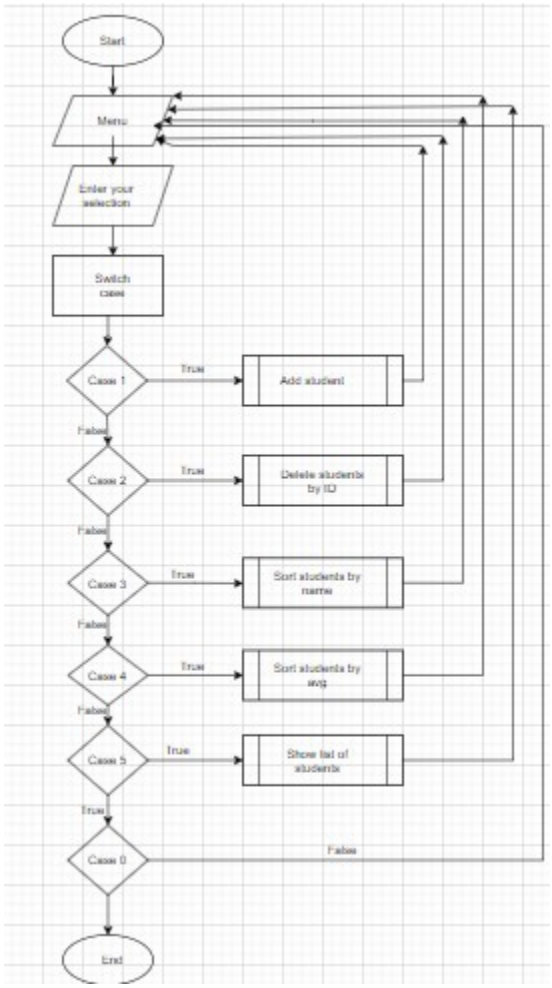


Image 3: Function

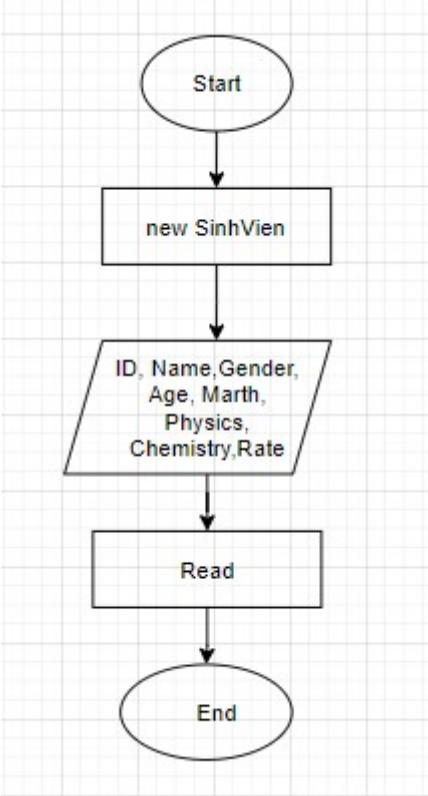


Image 4: Case 1

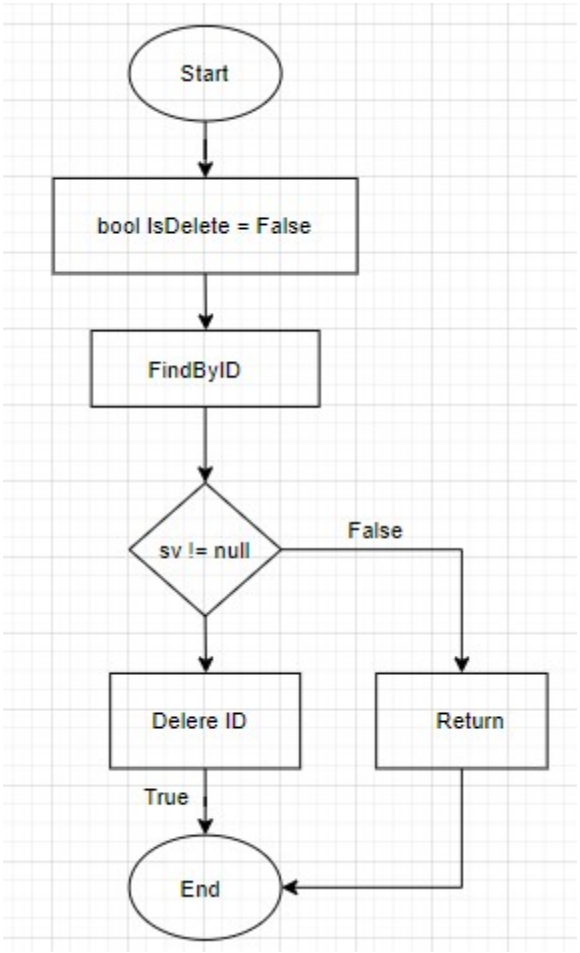


Image 5: Case 2

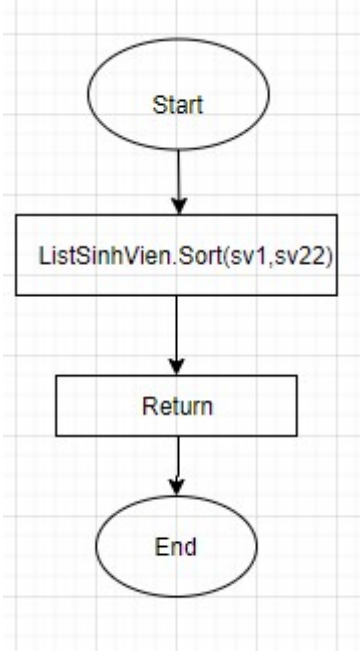


Image 6: Case 3

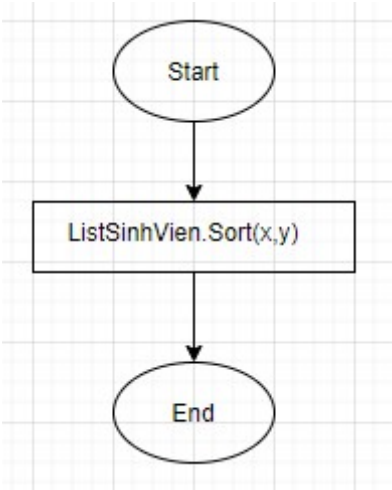


Image 7: Case 4

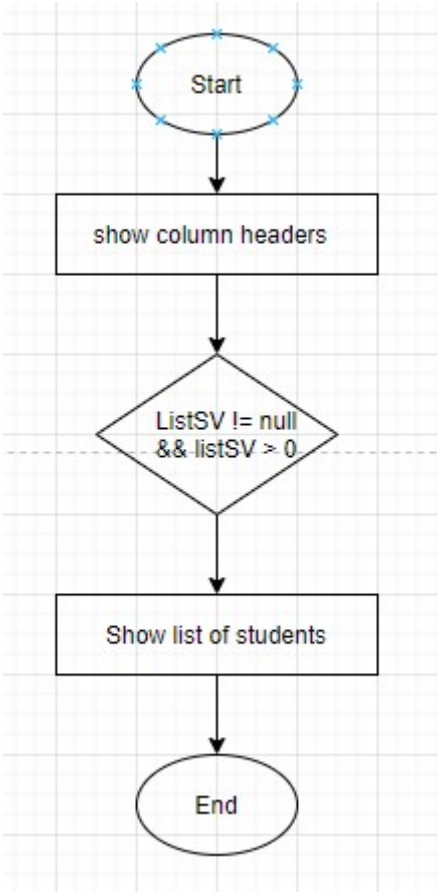


Image 8: Case 5

Task 3: Demonstrate the compilation and running of a program

I. Introduce how the problem is solved

- To demonstrate my basic programming and problem-solving skills. I used C# to create a student management application. First I will draw the flowchart. It helps me navigate how the program should work and I can determine what I need to do.

II. Source code and screen shots of the final result
II.1/ Source code

```
15 references
class SinhVien
{
    3 references
    public int ID { get; set; }
    4 references
    public string Name { get; set; }
    2 references
    public string Gender { get; set; }
    2 references
    public int Age { get; set; }
    3 references
    public double DiemToan { get; set; }
    3 references
    public double DiemLy { get; set; }
    3 references
    public double DiemHoa { get; set; }
    9 references
    public double DiemTB { get; set; }
    5 references
    public string HocLuc { get; set; }
}
```

- Class SinhVien: to define the student' properties such as: ID, name, age, math, chemistry, avg,…. Set takes the value in, get takes the value out.
  + int: Used to store student object with attributes like id, age, ect.

```
1 reference
public void NhapSinhVien()
{
    SinhVien sv = new SinhVien();
    Console.Write("ID: ");
    sv.ID = Convert.ToInt32(Console.ReadLine());

    Console.Write("Name: ");
    sv.Name = Convert.ToString(Console.ReadLine());

    Console.Write("Gender: ");
    sv.Gender = Convert.ToString(Console.ReadLine());

    Console.Write("Age: ");
    sv.Age = Convert.ToInt32(Console.ReadLine());

    Console.Write("Diem toan: ");
    sv.DiemToan = Convert.ToDouble(Console.ReadLine());

    Console.Write("Diem ly: ");
    sv.DiemLy = Convert.ToDouble(Console.ReadLine());

    Console.Write("Diem hoa: ");
    sv.DiemHoa = Convert.ToDouble(Console.ReadLine());

    TinhDTB(sv);
    XepLoaiHocLuc(sv);

    ListSinhVien.Add(sv);
}
```

- This function helps us enter student information.

```
1 reference
public void SortByDiemTB()
{
    ListSinhVien.Sort((x,y) => y.DiemTB.CompareTo(x.DiemTB));
    ListSinhVien.Sort((x, y) => x.DiemTB.CompareTo(y.DiemTB));
}

1 reference
private void TinhDTB(SinhVien sv)
{
    double DiemTB = (sv.DiemToan + sv.DiemLy + sv.DiemHoa) / 3;
    sv.DiemTB = Math.Round(DiemTB, 2, MidpointRounding.AwayFromZero);
}
```

- Help us to calculate the average and sort the data.

```
1 reference
private void XepLoaiHocLuc(SinhVien sv)
{
    if (sv.DiemTB >= 8)
    {
        sv.HocLuc = "Gioi";
    }
    else if (sv.DiemTB >= 6.5)
    {
        sv.HocLuc = "Kha";
    }
    else if (sv.DiemTB >= 5)
    {
        sv.HocLuc = "Trung Binh";
    }
    else
    {
        sv.HocLuc = "Yeu";
    }
}
```

- Used to make the right and wrong decisions when implementing algorithms.
- If the average score is greater than 8, the student is classified as excellent. Else if the average score is greater than 6.5, the student is graded as good. Else if the average score is greater than 5, the student is graded average. Else, the student was graded weakly.

```
1 reference
public SinhVien FindByID(int ID)
{
    SinhVien searchResult = null;
    if (ListSinhVien != null && ListSinhVien.Count > 0)
    {
        foreach (SinhVien sv in ListSinhVien)
        {
            if (sv.ID == ID)
            {
                searchResult = sv;
            }
        }
    }
    return searchResult;
}
1 reference
public bool DeleteById(int ID)
{
    bool IsDeleted = false;
    // tìm kiếm sinh viên theo ID
    SinhVien sv = FindByID(ID);
    if (sv != null)
    {
        IsDeleted = ListSinhVien.Remove(sv);
    }
    return IsDeleted;
}
```

- Find and delete ID
  + foreach: assign the value of the first element in the array to a temporary variable and execute the block inside the "foreach" loop.
  + bool: using to delete student's ID

```
1 reference
public void SortByName()
{
    ListSinhVien.Sort(delegate (SinhVien sv1, SinhVien sv2)
    {
        return sv1.Name.CompareTo(sv2.Name);
    });
}
```

- Sort students alphabetically.

```
3 references
public void ShowSinhVien(List<SinhVien> listSV)
{
    // hien thi tieu de cot
    Console.WriteLine("{0, -5} {1, -20} {2, -5} {3, 5} {4, 5} {5, 5} {6, 5} {7, 10} {8, 10}",
          "ID", "Name", "Gender", "Age", "Toan", "Ly", "Hoa", "Diem TB", "Hoc Luc");
    // hien thi danh sach sinh vien
    if (listSV != null && listSV.Count > 0)
    {
        foreach (SinhVien sv in listSV)
        {
            Console.WriteLine("{0, -5} {1, -20} {2, -5} {3, 5} {4, 5} {5, 5} {6, 5} {7, 10} {8, 10}",
                              sv.ID, sv.Name, sv.Gender, sv.Age, sv.DiemToan, sv.DiemLy, sv.DiemHoa,
                              sv.DiemTB, sv.HocLuc);
        }
    }
    Console.WriteLine();
}
```

- It will display the list of students entered in case 1.

```
3 references
public List<SinhVien> getListSinhVien()
{
    return ListSinhVien;
}

4 references
public int SoLuongSinhVien()
{
    int Count = 0;
    if (ListSinhVien != null)
    {
        Count = ListSinhVien.Count;
    }
    return Count;
}
```

- This will help us to check the number of students and retrieve the data from the menu.

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        QuanLySinhVien quanLySinhVien = new QuanLySinhVien();

        while (true)
        {
            Console.WriteLine("\n         CHUONG TRINH QUAN LY SINH VIEN           \n");
            Console.WriteLine("************************MENU************************");
            Console.WriteLine("**  1. Them sinh vien.                             **");
            Console.WriteLine("**  2. Xoa sinh vien theo ID.                      **");
            Console.WriteLine("**  3. Sap xep sinh vien theo ten.                 **");
            Console.WriteLine("**  4. Sap xep sinh vien theo diem trung binh.     **");
            Console.WriteLine("**  5. Hien thi danh sach sinh vien                **");
            Console.WriteLine("**  0. Thoat                                       **");
            Console.WriteLine("****************************************************");
            Console.Write("\n Nhap tuy chon: ");
            int key = Convert.ToInt32(Console.ReadLine());
            switch (key)
            {
                case 1:
                    Console.WriteLine("\n1. Them sinh vien.");
                    quanLySinhVien.NhapSinhVien();
                    Console.WriteLine("\nThem sinh vien thanh cong!");
                    break;
```

```
case 2:
    if (quanLySinhVien.SoLuongSinhVien() > 0)
    {
        int id;
        Console.WriteLine("\n2. Xoa sinh vien.");
        Console.Write("\nID: ");
        id = Convert.ToInt32(Console.ReadLine());
        if (quanLySinhVien.DeleteById(id))
        {
            Console.WriteLine("\nSinh vien co id = {0} da bi xoa.", id);
        }
    }
    else
    {
        Console.WriteLine("\nDanh sach sinh vien trong!");
    }
    break;

case 3:
    if (quanLySinhVien.SoLuongSinhVien() > 0)
    {
        Console.WriteLine("\n3. Sap xep sinh vien theo ten.");
        quanLySinhVien.SortByName();
        quanLySinhVien.ShowSinhVien(quanLySinhVien.getListSinhVien());
    }
    else
    {
        Console.WriteLine("\nDanh sach sinh vien trong!");
    }
    break;
```

```
case 4:
    if (quanLySinhVien.SoLuongSinhVien() > 0)
    {
        Console.WriteLine("\n4. Sap xep sinh vien theo diem trung binh.");
        quanLySinhVien.SortByDiemTB();
        quanLySinhVien.ShowSinhVien(quanLySinhVien.getListSinhVien());
    }
    else
    {
        Console.WriteLine("\nDanh sach sinh vien trong!");
    }
    break;

case 5:
    if (quanLySinhVien.SoLuongSinhVien() > 0)
    {
        Console.WriteLine("\n5. Hien thi danh sach sinh vien.");
        quanLySinhVien.ShowSinhVien(quanLySinhVien.getListSinhVien());
    }
    else
    {
        Console.WriteLine("\nDanh sach sinh vien trong!");
    }
    break;

case 0:
    Console.WriteLine("\nBan da chon thoat chuong trinh!");
    return;
default:
    Console.WriteLine("\nKhong co chuc nang nay!");
    Console.WriteLine("\nHay chon chuc nang trong hop menu.");
    break;
```

- The program runs the main function first, and then the menu will appear. Use a switch case to create a menu. In the menu, there are 6 cases.
- Case 1: Enter student information. When entering the correct student information, the student will be added successfully.
- Case 2: If want to delete student id. First, we have to enter the student ID that matches the one added in Case 1, and then it will print out "student with id = {0} removed". Else, it will print out "Empty Student List."
- In this case 0 acts as the exit of the program. If you choose another case that is not in the menu, the function will not be available and we can only select the function that is in the menu.

II.2/ Screen shots of the final result

- Menu:

```
        CHUONG TRINH QUAN LY SINH VIEN

**************************MENU*****************************
**  1. Them sinh vien.                                    **
**  2. Xoa sinh vien theo ID.                             **
**  3. Sap xep sinh vien theo ten.                        **
**  4. Sap xep sinh vien theo diem trung binh.            **
**  5. Hien thi danh sach sinh vien                       **
**  0. Thoat                                              **
************************************************************

 Nhap tuy chon:
```

- Option 1:

```
 Nhap tuy chon: 1

1. Them sinh vien.
ID: 200662
Name: Lih
Gender: Nu
Age: 19
Diem toan: 7
Diem ly: 8
Diem hoa: 9

Them sinh vien thanh cong!
```

```
 Nhap tuy chon: 1

1. Them sinh vien.
ID: 200123
Name: Ah
Gender: Nam
Age: 20
Diem toan: 7
Diem ly: 5
Diem hoa: 7

Them sinh vien thanh cong!
```

```
 Nhap tuy chon: 1

1. Them sinh vien.
ID: 200456
Name: Hah
Gender: Nu
Age: 19
Diem toan: 4
Diem ly: 5
Diem hoa: 6

Them sinh vien thanh cong!
```

- Option 2:

```
 Nhap tuy chon: 2

2. Xoa sinh vien.

ID: 200662

Sinh vien co id = 200662 da bi xoa.
```

- Option 3:

```
 Nhap tuy chon: 3

3. Sap xep sinh vien theo ten.
ID    Name                Gender   Age  Toan    Ly   Hoa    Diem TB    Hoc Luc
200123 Ah                  Nam      20     7     5     7       6.33 Trung Binh
200456 Hah                 Nu       19     4     5     6          5 Trung Binh
```

- Option 4:

```
 Nhap tuy chon: 4

4. Sap xep sinh vien theo diem trung binh.
ID    Name                Gender   Age  Toan    Ly   Hoa    Diem TB    Hoc Luc
200456 Hah                 Nu       19     4     5     6          5 Trung Binh
200123 Ah                  Nam      20     7     5     7       6.33 Trung Binh
```

- Option 5:

```
Nhap tuy chon: 5

5. Hien thi danh sach sinh vien.
ID    Name                Gender    Age   Toan    Ly    Hoa    Diem TB    Hoc Luc
200662 Lih                 Nu         19      7     8      9          8        Gioi
200123 Ah                  Nam        20      7     5      7       6.33 Trung Binh
200456 Hah                 Nu         19      4     5      6          5 Trung Binh
```

- Option 0:

```
Nhap tuy chon: 0

Ban da chon thoat chuong trinh!
```

III. Explain briefly what is Software Development Life Cycle

- SDLC is intended to describe the process of planning, developing, testing, and implementing an information system, and it is inexpensive to maintain and inexpensive to improve.(Rajkumar, 2021)

  + Tessting:
  - When the software is ready, it is sent to the testing department where Test team tests it thoroughly for different defects.

  + Design:
  - It has two steps:
    HLD – High-Level Design – It gives the architecture of the software product to be developed and is done by architects and senior developers
    LLD – Low-Level Design – It is done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

  + Analysic:
  - Once the requirement gathering and analysis is done the next step is to define and document the product requirements and get them approved by the customer. This is done through the SRS (Software Requirement Specification) document.

  + Deployment:
  - This is the phase where we start building the software and start writing the code for the product. The outcome from this phase is Source Code Document (SCD) and the developed product.

  + Requirement analysis:
  - Requirement gathering and analysis is the most important phase in the software development lifecycle.

  + Deployment & Maintenance:

- After successful testing, the product is delivered/deployed to the customer for their use. Deployment is done by the Deployment/Implementation engineers. Once when the customers start using the developed system then the actual problems will come up and needs to be solved from time to time. Fixing the issues found by the customer comes in the maintenance phase. 100% testing is not possible – because, the way testers test the product is different from the way customers use the product. Maintenance should be done as per SLA (Service Level Agreement)
(Rajkumar, 2021)

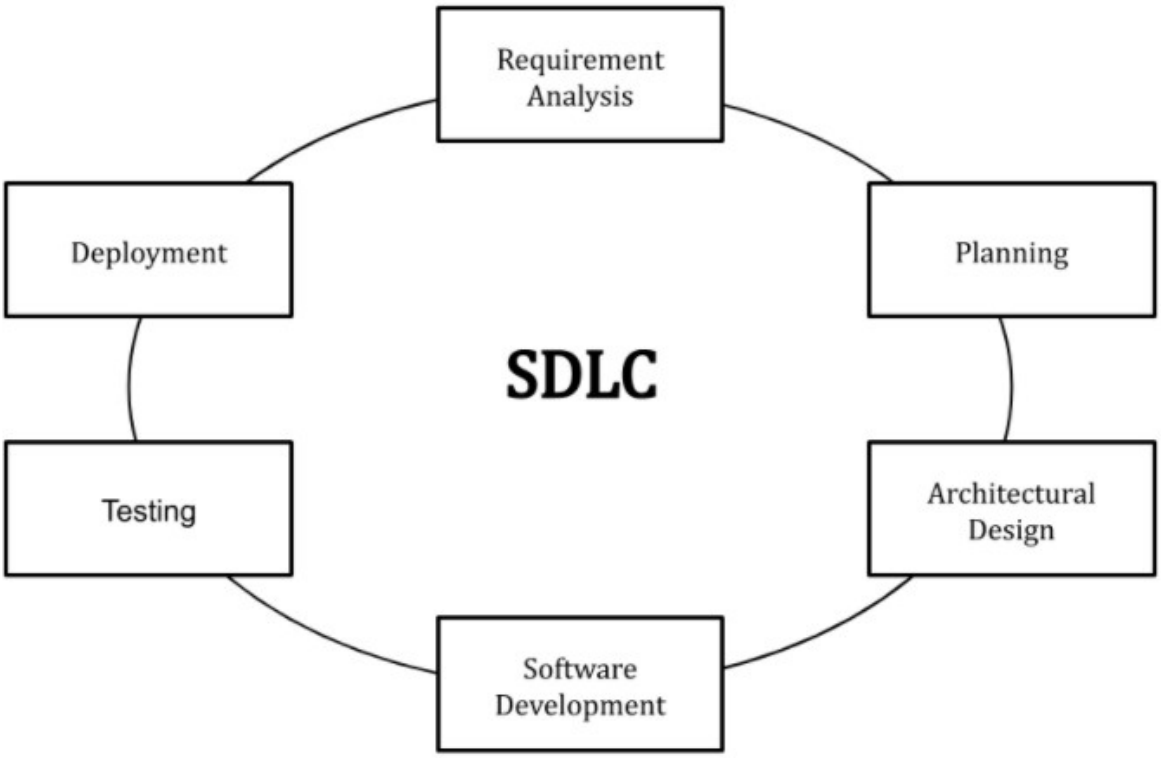


Image 9: Stages of the Software Development Life Cycle(source: Internet)

IV. Explain how the source code is compiled and run

- Source code is the fundamental component of a computer program that is created by a programmer. It can be read and easily understood by a human being. When a programmer types a sequence of C programming language statements into Windows Notepad, for example, and saves the sequence as a text file, the text file is said to contain the source code.(Scott Wallask, 2021)

- Step-by-step process of C# code compilation: (argha_c14, 2021)

  + **Step 1:** Write a C# code.

  + **Step 2:** Compile the code using a C# compiler.

  + **Step 3:** Now compiler checks if the code contains an error or not. If no error is found then the compiler move to the next step. Or if the compiler found errors, then it will immediately tell the developer that an error is found in the given line, so that the developer can correct them. After correcting the error when you again run the code the compiler again check for the errors, if no error found then it will move to the next step or if an error is found then the compiler gives a message to the developer. In C# there are two types of errors:

  Compiler error

  Runtime error

  + **Step 4:** Languages such as Java or C# are not directly converted or compiled into machine-level language or machine instructions. These languages need to be converted to an intermediate code first, which is a partially half compiled code. For C#, the source code is converted to an intermediate code which is known as **Common Intermediate Language (CIL)** or **Intermediate Language Code (ILC or IL code)**. This CIL or IL Code can run on any operating system because C# is a *Platform Independent* Language.

  + **Step 5:** After converting the C# source code to Common Intermediate Language (CIL) or Intermediate Language Code (ILC or IL code, the intermediate code needs to be converted to machine understandable code. C# uses the *.NET Framework* and as part of this .NET Framework, the *Virtual Machine component* manages the execution of programs written in any language that uses the . NET Framework. This virtual machine component is known as **Common Language Runtime (CLR)** which translates the CIL or IL code to native code or machine understandable code or machine instructions. This process is called the **Just-In-Time (JIT) Compilation** or **Dynamic Compilation** which is the way of compiling code during the execution of a program at run time only.

  + **Step 6:** Once the C# programs are compiled, they're physically packaged into **Assemblies**. An assembly is a file that contains one or more namespaces and classes. As the number of classes and namespaces in program grows, it is physically separated by related namespaces into separate assemblies. Assemblies typically have the file extension **.exe** or **.dll,** depending on whether they implement applications or

libraries respectively, where EXE stands for *Executable* and DLL stands for *Dynamic Link Library*. An EXE (Executable) file represents a program that can be executed and a DLL (Dynamic Link Library) file includes code (Eg: Library) that can be reused across different programs.

+ **Step 7:** Now, the C# compiler returns the output of the given c# code.

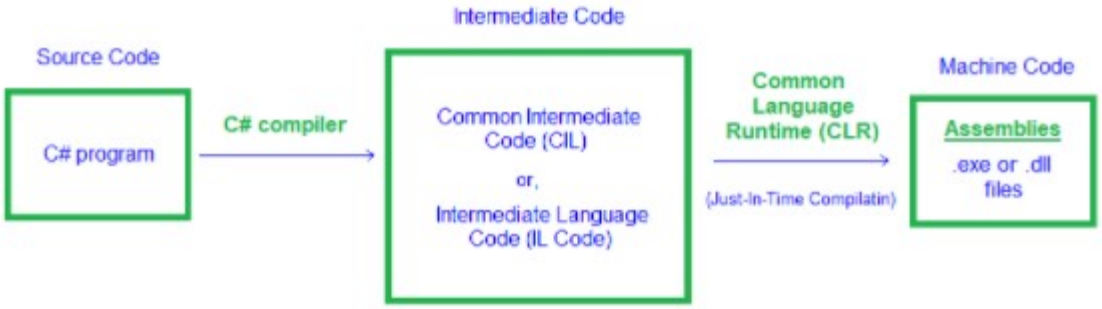


Image 10: Compilation process of C# code

Task 4: Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language.

I. Include Test cases

Test	Test Description	Test Steps	Test Data	Expected Results	Actural Results	Pass/False
1	Used to import student list.	1. Enter 2. Add student	ID: 200662 Name: Linh Gender: Female Age: 19 Math: 7 Physics: 8 Chemistry: 9	Add student	Successfull	Pass
2	Delete ID	1. Enter 2. Delete students by ID	ID:200662 ID:200456 ect.	Student with id 200662 has been deleted	Successful	Pass

3	Sort by name	1.Enter 2. Sort students by name	ID: 200662 Name: Linh Gender: Female Age: 19 Math: 7 Physics: 8 Chemistry: 9 Avg:8 Rate:Excellent	Show ID, Name,Gender, Age, rate, Avg,ect.	Successful	Pass
4	Sort of student by average score	1.Enter 2. Sort students by avg	ID: 200662 Name: Linh Gender: Female Age: 19 Math: 7 Physics: 8 Chemistry: 9 Avg:8 Rate:Excellent	Show average score	Successful	Pass
5	Export the list entered case	1.Enter 2. Show list of students	ID: 200662 Name: Linh Gender: Female Age: 19 Math: 7 Physics: 8 Chemistry: 9 Avg:8 Rate:Excellent	Show ID, Name,Gender, Age, rate, Avg,ect.	Successful	Pass
0	Exit	1.Enter 2.Exit	0	Exit program	Successfull	Pass

II. Evaluate how the problem is solved from the designed algorithm to the execution program written by a specific programming language

- In terms of advantages, the functions and algorithms are quite simple, and the program is not too complicated. The program runs very simply and fast.

- In terms of challenges, several interlocking structures and processes make program development complicated, and the app has some bugs that I can't fix.

Conclusion

I think my program is not very good. I need to change some places to be more suitable.

Through creating this student management program, I have learned a lot. It helped me improve my background knowledge.

In the future, I will write more complete student management software with many functions based on the designed algorithms.

I hope that the program I created can help teachers and schools manage students more effectively.In the future, I hope I can write a perfect student management program.

Bibliography

Kumari, R. (2021, December 19). Retrieved from analyticssteps.com: https://www.analyticssteps.com/blogs/what-algorithm-types-applications-characteristics?fbclid=IwAR2LcKdIqtven0LZ0L2RjJTXB5EAMmQKvp00IKHKjNoybNY2h0gyRdkRHGI

LearnComputerScience. (2021, 10 22). Retrieved from learncomputerscienceonline.com: https://www.learncomputerscienceonline.com/

Rajkumar. (2021, December 20). Retrieved from softwaretestingmaterial.com: https://www.softwaretestingmaterial.com/sdlc-software-development-life-cycle/

Thakur, A. (2021, December 19). Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/insertion-sort-in-chash?fbclid=IwAR0Eu8-JXkGZkVlPh-q2jwm4hBo1c82xtvg0bXkEIHoMO0Qe9xbTt58ibLc

Wallask, S. (2021, December 25). Retrieved from searchapparchitecture.techtarget.com: https://searchapparchitecture.techtarget.com/definition/source-code

Way2Benefits, C. (2021, December 19). Retrieved from way2benefits.com: http://way2benefits.com/advantages-disadvantages-algorithm/?fbclid=IwAR0wppU6u0urTGHMOc1yEdCLMqyC-tKa82PLUu_rv-nW6CiaaHtvWGmXz9s