Program Development Life Cycle

The Program Development Life Cycle is a set of phases and steps that are followed by developers to define, design, develop and maintain a computer program. Typical phases of the life cycle are:

- **Gather and Analyze the Program Requirements:** The developers must obtain the program requirements from the users and document the requirements. Typically, a standard form is used to develop the requirements.

- **Design the User Interface:** After understanding the requirements, the next step is to design the user interface. Typically a business information system consists of more than one program – it may have several forms, few reports, and other programs. The users interface defines all the input forms, output reports, programs, and menu system to organize all these objects.

- **Design the Program:** A business function when implemented in a computer system is termed as a program. Before a program is developed, it must be designed. Program developers use tools such as pseudocode, flow chart, and hierarchy chart to design programs.

- **Code the Program:** After a program is designed, it is implemented. A program code is a set of instructions developed by a programmer to carry out the business functionality in a particular language such as VB, C#, Java, ASP, etc.

- **Test the Program:** As the program is being coded, and after the code is completed it must be tested to see that program is running properly and it produces required outputs with appropriate input data.

- **Document the Program/System:** As a program is being coded, and after the coding is completed, the developers should document the program. Documenting a program means writing down some instruction for the users, the purpose of the program, the way it performs the tasks, the inputs and the outputs expected. Once each program or sub-program is documented, it is necessary to create documentation for the whole business information system.

- **Maintain the Program/System:** After the programs are developed and documented, it is placed into operation. During the operation, a program may fail to perform its objective and it might be necessary to add new functionality to a program or system. Changing program design, coding, and updating programs are part of maintenance.
Gather and Analyze the Program Requirements

Many business information systems fail, or never been used, due to poor requirements for the system to start with. If the business requirements are faulty, the design is going to be wrong, and the implementation or coding is going to be of course faulty, and the system will not perform its functionality. Thus it is important to gather information or requirements of a program or system. First step is to gather requirements (typically by a business analyst) from the business users through interviews, sending questionnaire, sampling existing documents and programs, and any other fact-finding methods. The second step is to document the requirements. Many companies use specific styles or tools to document the requirements. Below is an example of a requirements document of a business process program.

```
REQUIREMENTS DOCUMENT

Date submitted:      January 23, 2008
Application title:   Hotel Room Selection
Purpose:             The hotel room selection program will allow a user to view different room types and make a room selection.
Program Procedures:  From a window on the screen, the user should choose the room type and then make a room selection.
Algorithms, Processing, and Conditions:
1. The user must be able to view a standard room and a deluxe room until he or she makes a room selection.
2. When the user chooses a room type, a picture of that room type should appear in the window.
3. Only one picture should be displayed at a time, so if a user chooses a standard room, only the standard room picture should be displayed; if a user then chooses a deluxe room, the deluxe room picture should be displayed and the standard room picture should not be displayed.
4. When the user makes a room selection, a confirming message should be displayed. In addition, the user should be prevented from identifying a room type after the room selection is made.
5. After the user makes a room selection, the only allowable action is to exit the window.

Notes and Restrictions:
1. The user should not be able to make a room selection until he or she has chosen a room type.

Comments:
1. The pictures shown in the window should be selected from the pictures available on the Web.
```
Designing the User Interface

A business information system typically consists of three layers: the presentation layer, the processing layer, and the data layer as depicted here. The presentation layer contains the objects that are interfaced by the user such as forms, scanners, touch-screens, printouts, statements, reports, etc. The processing layer consists of programs that are called by these objects to perform the business processes required by the users. The data layer contains data that are already stored in files or databases or need to be stored. A user only sees the presentation layer and so the name; the programming and data details are hidden from the user. In earlier days, a programmer typically used to write codes, but in today's business environments, a programmer needs to know about all three layers to successfully develop an information system. When developing an information system using Graphical User Interface (GUI), the Presentation Layer and the Processing Layer cannot be separated – rather, there is a design view and a code view – and together they are termed as the Top Layer. The file or database is termed as the Bottom Layer.

From the requirements documents, typically a systems analyst defines the various types of inputs, outputs, programs, menus, files, and databases required for the system. Next step is to define the details of data that go into each of these objects. For example, a set of data that might go into a customer form might include customer's last name, first name, address, city, state, zip code, telephone number, e-mail address, etc. For a program, it might include the steps in simple English language that the program must perform to achieve the business objective. For a file or database, it is necessary to define the fields and types of data that each field must store in a file or table of a database. The next step is to design each object. Designing means how an object will looks like before it is being developed - just like an architect shows design layouts of a building on papers before constructing with raw materials.

For an information system, we need to design forms, reports, programs, menus, files, tables, etc. For a form or report, the design layout mean how form or report will look like, for a program design, a programmer needs to show the steps of a business process that satisfies the user requirements as well as facilitates the coding process. We will mainly focus on the program design in this class.

A program typically takes inputs, performs some processing depending on business rules and then produces some outputs. Designing a program means developing algorithms for these functionalities. Commonly, three tools are available that facilitate the program design: pseudocode, flowchart, and hierarchy chart.