Information Systems and Programming Basics

Foundational Curriculum
Cluster 3: ICT Process
Module 5: General HIT Knowledge/System Use
Unit 1: Information Systems and Programming Basics
FC-C3M5U1

Curriculum Developers: Angelique Blake, Rachelle Blake, Pauliina Hulkkonen, Sonja Huotari, Milla Jauhiainen, Johanna Tolonen, and Alpo Värri

14/60

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Unit Objectives

- Identify basic computer system/application programming concepts, describe the role of programming in a health IT/eHealth system, and possess familiarity with the fundamental capabilities of computers
- Explain basic computer science principles and methods
- Describe basic computer system programming, and identify common programming languages and applications
- Review the importance of usability engineering, graphical user interface evaluations, cognitive aspects of information processing, user accessibility, etc.
- Describe the use of, and explain how to support others in the use of, organization-supported software
- Review common computer terminology (e.g., program, operating system) and

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

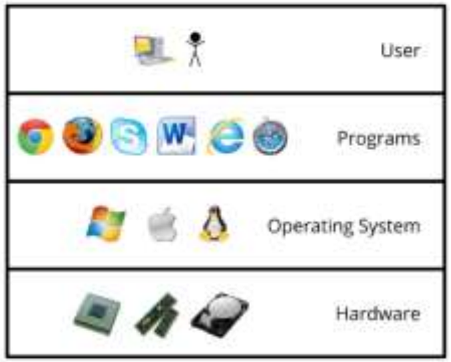
Basic Computer Science Concepts

- A **computer** is an electronic device that accepts data as input, processes data into information, stores information for future uses, and outputs the information whenever it is needed.
- **Hardware** is the material you can touch, as opposed to software which is abstract and exists only in a virtual world as computer code.
- **Software** includes all computer programs regardless of their architecture; for example, executable files, libraries and scripts
  - However, software and hardware share mutual properties: software consists of clearly-defined instructions that upon execution, instructs hardware to perform the tasks for which it is designed.

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

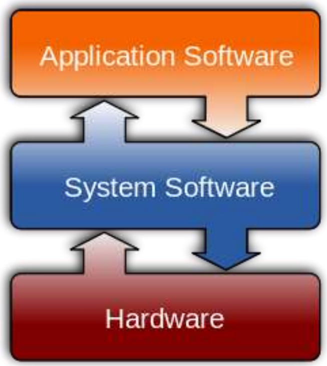
The Basic Computer System

- The basic computer system consists of **hardware** and **software**
  - Hardware is made up of physical devices like tablets, including hard drives, central processing units (CPUs), speakers, and **peripherals** such as servers, cables, printers, mice and styluses.
  - Software is made up of intangible elements such as the programs or internal systems housed within the hardware, such as the operating system (OS), random access memory (RAM) and data or information made up of bytes (now usually stored in increments of gigabytes or terabytes), files and applications.
    - A computer **program** is defined as a collection of instructions that performs a specific task when executed by a computer. A program may also be called an **application**.



Your computer as a layered system

- Basic computer system and application programing is a process that leads from formulating an original computing problem to executing a software program.
  - When done at the computer level, it is called “computer programming”. When done with a specific sub-function within a computer, it is called “application programming”.



This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

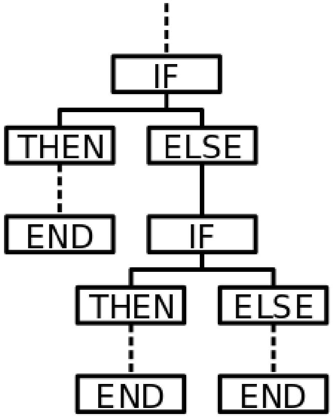
Fundamental Capabilities of Computers

- **What are the fundamental capabilities of computers?**
  - As stated before, a **computer** is a programmable, electronic device that accepts data, performs operations on that data, presents the results, and stores the data or results as needed for user retrieval and use
  - A computer is capable of performing four general operations:
    1. **Input**: Users input data into computers. Users include programmers, analysts, engineers, and end users (clients, customers or, in the case of healthcare environments, eHealth workers).
    2. **Processing**: The computer transforms, computes, and processes, raw data into usable information.
    3. **Output**: The information can be displayed as a lines of text, a document, graphic, photo, multimedia presentation or other content.
    4. **Storage**: The information can be saved and retrieved for later viewing and use, either internally or on an external device.

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Programming and Coding

- **Programming** involves activities such as analysis, developing understanding, generating algorithms, verification of requirements of algorithms, determining their correctness and resources consumption, and implementation
- Computer **coding** is using a system of signals (most basically, zeros and ones, or **binary code**) to represent letters or numbers in transmitting messages. These messages can become instructions, and when they are written by a programmer in a programming language they are often called **source code.**
  - Implementation is commonly referred to as coding of algorithms in a target programming language
  - The purpose of programming is to find a sequence of instructions that will automate performing a specific task or solving a given problem. Programming thus often requires expertise in many different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.
- Programming plays an important role in a health IT/eHealth system



This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Basic Computer Science Principles and Methods

- On the following slides, you will become aware of the following basic computer science principles and methods, and describe and identify the basic computing topics, including:
  - programming languages
  - software engineering/basic computer architecture methods
  - data structures
  - database management systems and
  - other information and communications technology concepts

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Programming Languages

- Following are some common examples of languages used in health data programming:
  - Multipurpose/scientific programming:
    - ALGOL, BASIC, COBOL, FORTRAN, LISP, MAINSAIL, MUMPS, PASCAL, PL/I, SIMULA
  - Systems programming:
    - BCPL, BLISS, C++, PL/M, MUMPS
  - **HL7** (health level 7)
  - **SQL** (not an actual programming language, but a database management language for relational databases)

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Programming Languages (cont’d)

- Programming languages can be divided to two categories: low and high level languages
- Low level:
  - Machine code: zeros and ones 011011 (1st Gen.)
  - Assembly language with words describing commands (2nd Gen)
- High level:
  - Fortran, C, Java: symbols in commands (3rd Gen)
  - SQL: more complex commands (4th Gen)
  - Prolog: visual programming (5th Gen)

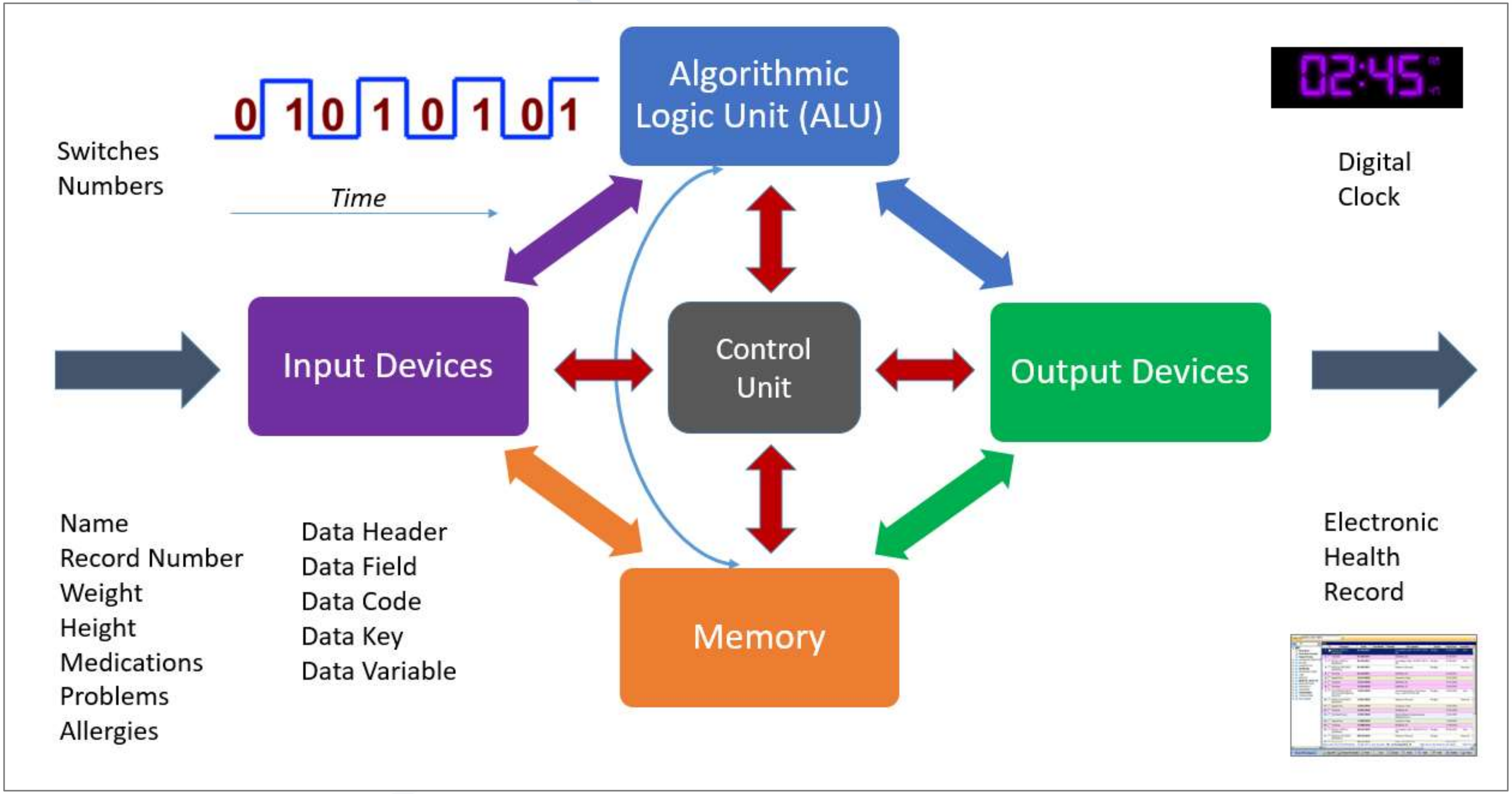
This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Programming Principles and Software

- A common form of software is a program. A **program**, e.g., word processing, internet browser, or media player program, is a specific collection of instructions that perform a specific task using a programming language
  - Different programming languages are used for different purposes
    - Some are more visual than others; some are very mathematical
  - Programs can be small and simple, or very large and complex
- Software is usually written in high-level programming languages that are easier and more efficient for humans to use (closer to natural language) than machine language.
  - High-level languages are compiled or interpreted into machine language object code.

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

The Basics of Computer Architecture



Switches
Numbers

Digital
Clock

Name
Record Number
Weight
Height
Medications
Problems
Allergies

Data Header
Data Field
Data Code
Data Key
Data Variable

Electronic
Health
Record

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Algorithms

- **Algorithm**
  - A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. It is a procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation.
- Computer programs implement different algorithms, but algorithms have been used long before computers existed
- Algorithms can be written in pseudocode or with flowcharts
  - **Pseudocode** is an informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading.
  - **Flowcharts** are ways to represent diagrams of workflow, as you learned about in Module 2. A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows.
    - They are used in analyzing, designing, documenting or managing a process or program. An algorithm is a step-by-step analysis of the process, while a flowchart explains the steps of a program in a graphical way
    - Examples are on the next slide

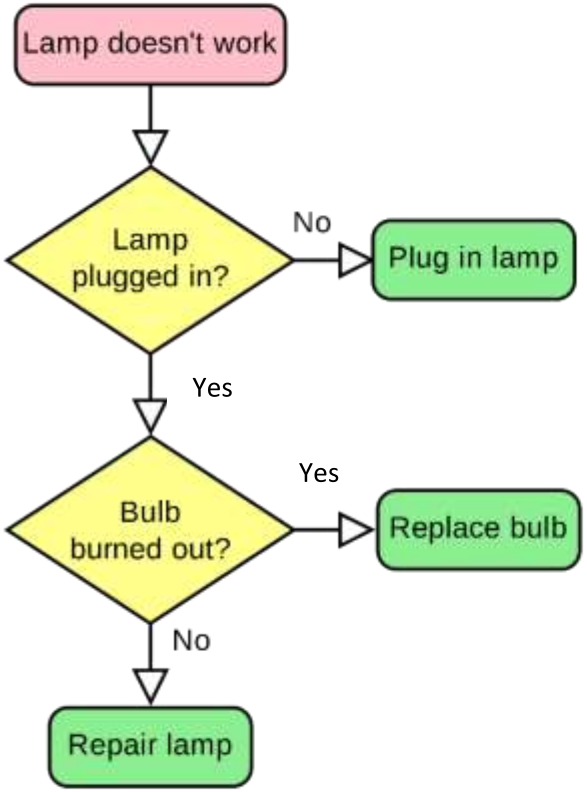
This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Pseudocode & Flowchart example

Fig 1: Pseudocode

```
While Lamp = 0 %lamp doesn't work
       If Plug = 0
              Plug = 1  %Plug lamp
       else
              if Bulb = 0
                     Bulb = 1  %replace bulb
              else
                     Lamp = 1  %repair lamp
              end
       end
end
```

Fig 2: Flowchart



This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Software Development

- **Software development** is the conception, development, and final distribution of software. As it is a fliud invention, it is subject to updates, changes, new versions, and user modifications. Specifically in healthcare, end users need a flexible software in which to access, retrieve, and store data
- Steps of Software Development:
  - Research
  - Prototype
  - Design and plan program
  - Write program or code
  - Modification
  - Respond to needs of in-users
  - Testing and Beta Testing for functionality
  - Support and Maintenance: safety and efficacy
  - Updates

```
 * @var boolean
 */
define('PSI_INTERNAL_XML', false);
if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work
        properly.");
}
require_once APP_ROOT.'/includes/autoloader.inc.php';
// Load configuration
require_once APP_ROOT.'/config.php';
if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
}
                                          strtolower(
```

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

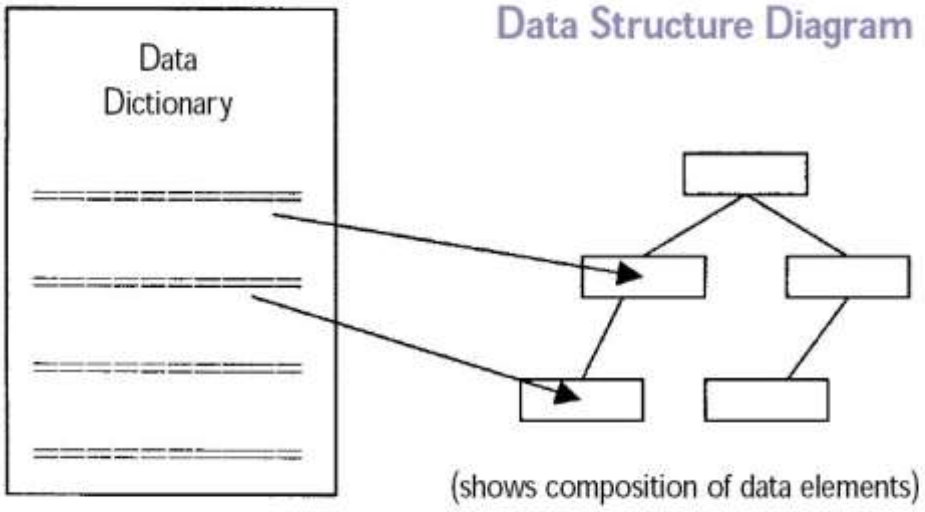
Prototyping

- Before writing an actual final product, it is good to build a prototype to explain features. The three main types of prototypes are:
  - Paper prototype
  - Electronic prototype (without functionalities)
  - Functional program with some limitations
- Prototyping is a good tool for involving stakeholders in the product development
  - It is important to ask healthcare workforce workers to test prototypes and give feedback e.g., about usability
  - This is an important phase where GUI (graphical user interface) can be tested and validated: how does the user interact with the screens, the devices, the functionality, the ergonomics, the ”look and feel” of the integrated software and hardware

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Data Structures

- A **data structure** is a particular way of organizing data in a computer so that it can be used efficiently



Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services.

- Often efficient data structures are key to designing efficient algorithms
- Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

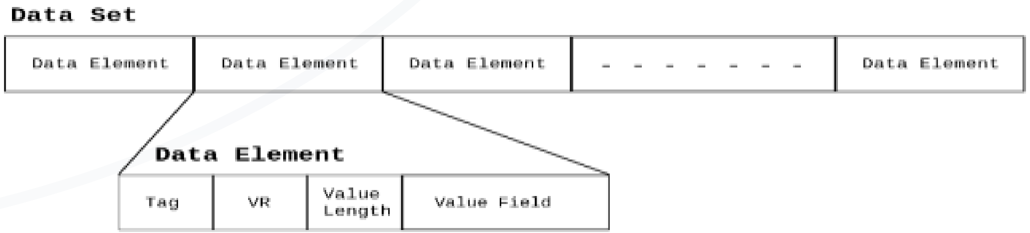
Data Structures (cont'd)

- There are several types of data structures used in healthcare:
  - A **data dictionary** describes the contents, format, and data structure of the database and the relationship between its elements, used to control access to and manipulation of the database
  - An **array** is a list of data, typically all of the same type. Each piece of data in an array is identified by an index number representing its position in the array. Arrays are zero based, which means that the first element in the array is [0], the second element is [1], and so on. Elements are accessed using an integer index to specify which element is required.
  - A **linked list** is a linear collection of data elements of any type, called nodes, where each node has a value and points to the next node in the linked list.

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Data Structures (cont’d)

- A **record** is an aggregate data structure. A record is a value that contains other values, typically in fixed number and sequence and typically indexed by names. The **data elements** of records are usually called **fields** or members. A **data set** contains multiple data elements.
- A **union** is a data structure that specifies which of a number of permitted primitive types may be stored in its instances, e.g. float or long integer.
- A **variant** (also called a tagged union, variant record, discriminated union, or disjoint union) contains an additional field indicating its current type, for enhanced safety.
- A **class** is a data structure that contains data fields, like a record, as well as various methods which operate on the contents of the record.



Data Set

Data Element	Data Element	Data Element	- - - - - - -	Data Element

Data Element

Tag	VR	Value Length	Value Field

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Data Structures (cont'd)

- Data structures are used for storing multiple data pieces together
- Example of Data Element, Data Array, and Specific Data Member
  - exampoints1= 95.0 (double variable, points for one student)
  - exampoints = [95.0; 50.5; 77.0; 45.0; 100.0] (stores points for five students)
  - Information can be searched: exampoints(3) = 77.0

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Data Hierarchy

- A **data field** holds a single fact or attribute of an entity. Consider a date field, e.g. "19 September 2018". This can be treated as a single date field (e.g. birthdate), or three fields, namely, day of month, month and year.
  - In terms of data storage, data fields are made of bytes and these in turn are made up of bits. Eight bits equal one byte.
- A **record** is a collection of related fields. A patient record may contain a name field(s), medical record number field, birthdate field, and so on.
- A **file** is a collection of related records. If a doctor has 40 patients on his admission list, then each patient would have a record (e.g. called Patient XX Admit Record) and the collection of 40 such records would constitute a file (in this case, called Dr. Doktor Patient Admit Record file).
- Files are integrated into a **database**. This is done using a Database Management System.

Hierarchy	Example
Database	**Patient Database** Admitted Patients Emergency Patients Outpatients
File	**Admitted Patient File** Demographics Record Health Maintenance Record Health Record
Record	**Health Record** Name MR Number Allergies Medications Problem List
Field	**Allergies** sulfa
Byte	01110011 Letter “s” (115 in ASCII)
Bit	0

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

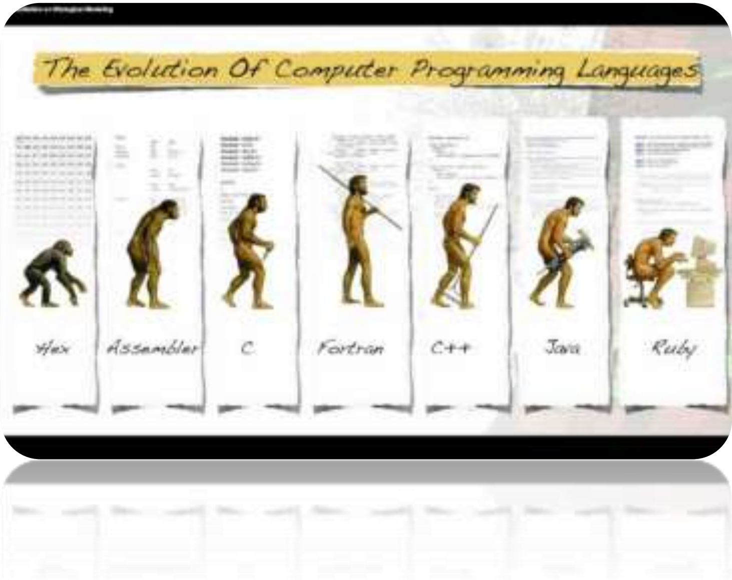
Database Management Systems

- A **database** is an organized collection of data.
- A **relational database**, on the other hand, is a collection of schemas, tables, queries, reports, views, and other elements, often used in healthcare. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as modelling the way a patient record is displayed or data is entered in a way that supports providers entering data or viewing results.
- A **database management system** (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.
  - A database is not generally portable across different DBMSs, but different DBMSs can become **interoperable** by using standards such as HL7 to allow a single EMR to work with another.

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

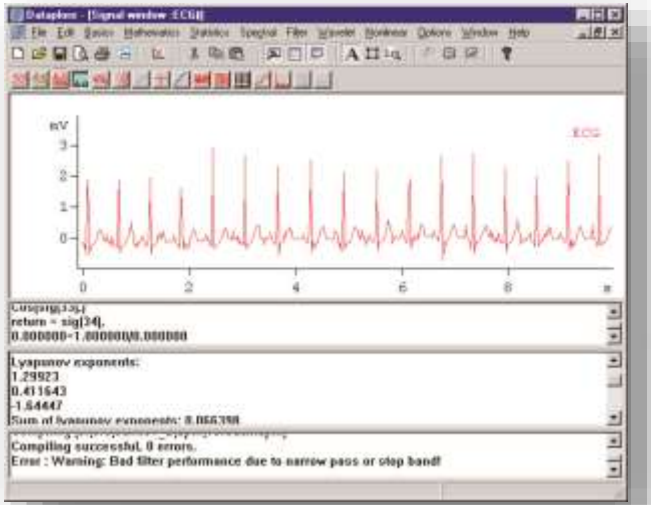
First, Second, and Third Generation Programming Languages

- First Generation: **Binary Code or Numerical Machine Code**
  - Computers do tasks based on binary code, a set of zeros and ones (01100001)
  - Unique sequences, applications are a set of these instructions
- Second Generation: **Assembly Language**
  - Assembly language uses words to represent instructions
  - Assembly code is translated into machine code for application
  - ”if” ”or” ”end” ”start”
- Third Generation: **High Level General Purpose Language**
  - Operations to programming are not unique to one computer anymore
  - Portable programs available, same code can be used in any computer
  - More languages were generated: Fortran, C, C++, Java, HL7



This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Programming in Healthcare Settings


Dataplore - [Signal window :ECG]
File Edit Basics Mathematics Statistics Spectral Filter Wavelet Nonlinear Options Window Help
mV
3
2
1
0
ECG
0
2
4
6
8
s

```
return = sig[34];
0.000000-1.000000/0.000000
```

```
Lyapunov exponents:
1.29923
0.411643
-1.64447
Sum of lyapunov exponents: 0.066398
```

```
Compiling successful, 0 errors.
Error : Warning: Bad filter performance due to narrow pass or stop band!
```

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Programming in Healthcare Settings

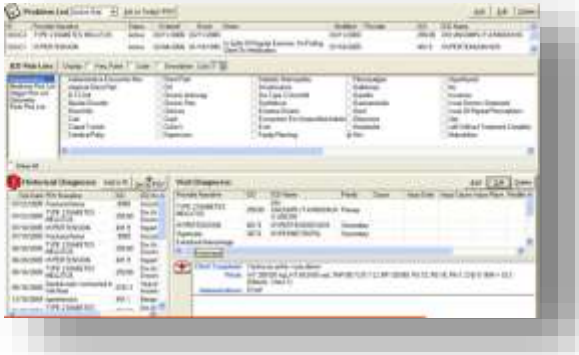
- Programming is an invisible but very important part of healthcare products
- On the previous slide some modern medical devices were presented. All are now digital, connected and/or including a computer in the workflow
- Settings for radiotherapy can be computed and stored, after the first treatment the patient setup is easier, as the settings are in the computer memory
- Monitors process signals before visualization
- Devices can be controlled via PC (MRI for example)

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

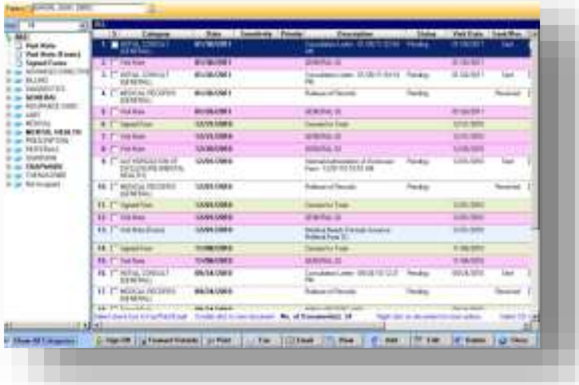
Usability Engineering

- When health information systems are deployed, they need to fit clinical workflow and be usable by a variety of end users
  - This includes healthcare workers, clinicians and even patients in some cases
- There is growing evidence that poorly designed user interfaces may actually contribute to medical errors*
- Therefore, usability engineering is paramount to design of eHealth technology and information systems, including EHRs
- User A and User B prefer different **graphical user interfaces (GUI),** a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators. It all depends on the individual preference

GUI



Better This?



Or, Better This?

**Kushniruk A., Monkman H., Borycki E., Kannry J. (2015) User-Centered Design and Evaluation of Clinical Information Systems: A Usability Engineering Perspective. In: Patel V., Kannampallil T., Kaufman D. (eds) Cognitive Informatics for Biomedicine. Health Informatics. Springer, Cham*

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Graphical User Interface Evaluations and User Accessibility

- Usability engineering and user-centered design principles can be applied together to improve the usability of clinical information systems. Approaches for achieving this include:
  - Employing **graphical user interface evaluations** at all phases of EMR implementation, including during design, build, testing, training and activation
  - Applying laboratory style usability testing
  - Using clinical simulations conducted in-situ and testing systems in real-world clinical settings
  - Utilizing rapid usability engineering methods throughout the design and implementation cycle of clinical information systems

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Cognitive Aspects of Information Processing

- Programmers should keep in mind the various human cognitive characteristics of cognitive development and cognitive style, when selecting different programming language paradigms, including procedural, object-oriented, visual, and script.*
  - This relates to coding and formulating queries in a program such as: would patients react better to an HTML form with fields or a complex PDF page where they need to enter complex information?; would nurses react better to radio buttons or check boxes in indicating drains present?
- Proving access to patient and consumer-facing information and technology that is accessible for a variety of disabilities
- It's more than just making systems "user-friendly", it's making them meaningfully usable

Journal of Information Systems Education, Vol 13(1) 59 A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmers' Cognitive Characteristics Garry L. White, Ph. D. 2002

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Programming in Healthcare Settings

- Programmers and IT specialists in the healthcare field need to understand the specialties in healthcare products:
  - Safety and accurate functions are crucial to maintain patient safety
  - Margins for error are very small or simply do not exist when it comes to patient care
  - Usability of software is important: as healthcare professionals are not specialists in IT or computers, the software should be intuitive, and updates should not change user interface remarkably

- When using software, healthcare professionals should be involved in the design and should be able to give as accurate feedback in redesign and optimization as possible, in order to improve the next generation software
- If some programs seem difficult or unintuitive to use, give feedback to the developer/IT-helpdesk, and explain WHY something is not working
- Work with your informatics team, if you have one, and if not, convey the message to your IT team
- Remember, the IT analyst is likely not a specialist in healthcare, thus s/he might not realize the special design demands and constraints in eHealth information systems and technology that allow full usability

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

The Use of Organization-Supported Software

- First and foremost, become aware of the policies and procedures related to your organization-supported software and programs
  - As every organization has its own software and methods, it is not possible to teach every system during education
- However, basic computing skills help in using healthcare software
  - Become familiar with basic programs like your organization's EMR and word processors, spreadsheets, text editors, single-sign-on programs, etc.
- In addition, Windows/Mac devices, common software programs, and web browsers, have a lot of instructions that are available on the Internet; use keywords to find what you are looking for, such as:
  - ”Windows 10 change power settings”
  - Print to file using PDF
- Participate in education and training for new software, ask questions and take notes
- Remember to follow your organization's guidelines and procedures at all times

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Unit Review Checklist

- ❑ Identified basic computer system/application programming concepts, described the role of programming in a health IT/eHealth system, and have familiarity with the fundamental capabilities of computers (EL05)
- ❑ Explained basic computer science principles and methods (GGB03)
- ❑ Described basic computer system programming, and identify common programming languages and applications (GGL01)
- ❑ Reviewed the importance of usability engineering, graphical user interface evaluations, cognitive aspects of information processing, user accessibility, etc. (GGL02)
- ❑ Described the use of, and explained how to support others in the use of, organization-supported software (GGB05)
- ❑ Reviewed common computer terminology (e.g., program, operating system) (GGB06)

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Unit Review Exercise/Activity

Find examples on the Internet of these types of data structures used in healthcare. Use the definitions supplied in this unit for reference.

- A healthcare data set
- Two or more examples of a record within that data set
- Two or more examples of a data field within that data set

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Unit Exam

1. Which of these are an example of first generation computer programming language?
    a. Binary Code
    b. An assembly language that uses words to represent instructions, translated into machine code for application
    c. High Level General Purpose Language
    d. None of the above

2. Which of these are an example of second generation computer programming language?
    a. Assembly Language
    b. Numerical Machine Code
    c. Languages such as Java, or HL7
    d. None of the above

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK

Unit Exam (cont’d)

3. Which of these are an example of third generation computer programming language?
   a. 01100001
   b. Languages such as Fortran, C, or C++
   c. ”if” ”or” ”end” ”start”
   d. All of the above

4. Which of the following is not an accurate definition of hardware and software?
   a. Hardware is the material you can touch and software is abstract
   b. Hardware is made of software and software is a specific set of programs
   c. Hardware is made up of physical devices like tablets and software only exists in the digital world
   d. Hardware includes printers, cables, and screens while software includes files and applications

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Unit Exam (cont'd)

5. Which is the order of general operations for a computer?
   a. Storage, input, processing, output
   b. Input, storage, processing, output
   c. Input, processing, output, storage
   d. Input, processing, storage, output

6. **Fill in the Blank**: Computer _________ is using a system of signals (most basically, _________ code, or zeros and ones,) which is used to represent letters or numbers in transmitting a message.

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552
EUUSEHEALTHWORK

Unit Exam

7. What does GUI stand for and what is its significance in clinical systems?

   a. General user input; the input entered into programs to be standardized by programmers

   b. General user input; the input processed into output by all users in C++

   c. Graphical user interface; to integrate new users graphics

   d. Graphical user interface; to improve the usability of clinical information systems

8. What is this an example of? *Select a date:* __.__.__ *to* __.__.__ *(09.09.1980 to 09.10.1980)*

   a. A data field

   b. A date field

   c. Both A. and B.

   d. None of the above

This work is produced by the EU*US eHealth Work Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 727552 EUUSEHEALTHWORK