CMPT 120

Topic: Introduction to Computing Science and Programming + Algorithm
Last Lecture

• i-clicker Practice Session
• Computing Science -> Problem Solving
• Steps in Problem Solving Process

Problem Solving Process:
1. State the problem
   • Figure out what the problem is and make sure we understand it
2. Design possible solutions
3. Select the “best” solution
4. Carry on the selected solution
5. Testing: has the selected solution solved the initial problem?

• Decomposition
  • Decompose the description of our solution into “finer” details
    • Example: *Buy 2 liters of organic 2% cow milk from Avalon*
Learning outcomes

At the end of this lecture, a student will be able to:

• Describe fundamental concepts pertaining to computing science
  • Algorithm
  • Computer program

At the end of this lecture, a student will have a better idea of what it means to solve a problem using a computer:

• State the problem
• Design an algorithm + analyze the data required
• Create a program
• Test a program
Today’s Menu

• Course Title: “Introduction to Computing Science and Programming 1”

• Illustrate the Problem Solving Process using a computer
  • Experience what it means to create a computer program
What is Programming?

• a.k.a. Software Development

• **Definition:**
  • Science and art of creating computer programs for the purpose of solving problems
Problem Solving Process using computers

1. State the problem
   - Figure out what the problem is and make sure we understand it

2. a) Design possible solution(s)
   - Solution is expressed as an algorithm

   b) Identify data
   - Input – identify data needed in order to solve problem
   - Structure the data and represent it in solution to problem
   - Output – identify data produced by solution to problem

3. Select the “best” solution
   - By analyzing algorithms
   - Which one is the most effective/efficient?

4. Implement the selected solution
   - We implement the algorithm into a computer program

5. Testing
   - Does the program execute?
   - Does it solve the problem?
Activity: Let’s solve another problem!

1. State the problem
   • Figure out what the problem is and make sure we understand it

Problem Statement:
Activity: Let’s solve another problem!

2.  a) Design possible solution(s)
    • Solution is expressed as an algorithm

   b) Identify data
    • Input – identify data needed in order to solve problem
    • Structure the data and represent it in solution to problem
    • Output – identify data produced by solution to problem
Algorithm - Definition

- A **finite sequenced** set of **unambiguous** steps that, once executed, produces a **result**
  
  - **finite**: This set of steps executes in a finite amount of time i.e. it should finish at some point
  - **sequenced**: The steps must be executed in the order in which they are listed
  - **unambiguous**: Each step is clear
  - **result**: This result solves the initial problem
More about algorithm

• Please, see the video “Algorithms from Khan Academy” listed under Resources on our home course web page
How do we express an algorithm?

1. Use a natural language like English
   - Example:
     - Problem Statement: Let’s figure out our course final grade
       - Take all the grades we obtained in course activities (e.g., assignments, exams, etc…)
       - Compute each grade as a % of final grade
       - Total them and the sum is our final grade out of 100%
How do we express an algorithm?

2. Use a mix of natural language and computer language - > **pseudocode**
   - **Example:**
     - **Problem Statement:** Let’s figure out our course final grade
     - Set finalGrade to 0
     - For each grade
       - Input grade
       - Input gradedOutOf (the grade’s maximum - out of)
       - Input percentOfFinalGrade (the grade’s % of the final grade)
       - Compute newGrade -> (grade * percentOfFinalGrade) / gradedOutOf
       - finalGrade = finalGrade + newGrade (keep a running total)
     - Print finalGrade
How do we express an algorithm?

3. Use a flowchart

Legend:
- Start and end of algorithm
- Processing
- Input or Output
- Decision
- Flow of execution

Start
Set finalGrade to 0

All grades read yet?

Yes

End
Print finalGrade

No

Input grade
Input gradedOutOf
Input percentOfFinalGrade
Compute newGrade
Keep running total for finalGrade
How do we express an algorithm?

4. In a diagram
   - Please, see "Here is an awesome way of expressing an algorithm (Way #4)" listed under Resources on our home course web page

5. In a video (verbal instructions and pictures)
   - Please, see "Here is another awesome way of expressing an algorithm (Way #5)" listed under Resources on our home course web page
Why do we need algorithms?
Back to our activity!

3. Select the “best” solution
   • We select the “best” solution by analyzing algorithms
   • Which one is the most effective/efficient?

4. Implement the selected solution
   • We implement the algorithm into a computer program

5. Testing
   • Does the program execute?
   • Does it solve the problem?
Computer Program

• **Definition**: …

• Before coming up with a definition, let’s play a game
  • Let’s see what it means to “Implement algorithm into computer program” -> Step 4!
Warning!

• First of all, let’s be very clear about computers: Computers are ___________ and will only do what we tell them to do!

• So we need to give the computer ___________________________________________ in order for it to do what we want it to do

• However, computers are very good at executing:
  • Very basic instructions e.g. calculations
  • Instructions very fast
  • Instructions on a very large amount of data
Activity: Let’s play at “programming” a computer!

1. We as human beings
   - Grab a piece of paper and draw a ___________

2. We as computers
   - Imagine we are computers, what instructions would we need in order to draw a ___________
So, what is a Computer Program?

- **Definition:**
  - A *finite sequenced* set of *computer programming language instructions*
  - These instructions tell the computer what to do
Summary

• Computing Science -> Problem solving
• Programming (software development)
  • Problem Solving using a computer
• Algorithm
  • How to express them -> 5 ways
  • Why do we need them
• Computer program
Next Lecture

- Definition of Programming language
- Evolution of programming languages
- Compiled versus interpreted programming languages
- Continue illustrate the Problem Solving Process

State problem \rightarrow Design algorithm + Identify data \rightarrow Implement algorithm into computer program \rightarrow Test it