

Fundamentals of Programming

Laboratory 1

Michał Bujacz

bujaczm@p.lodz.pl

Building B9 „Lodex” 207

Wednesdays 10:00,

Friday 13:00



Task points

10 laboratory classes (3 points each)

- 3h each (usually 1 hour tutorial, 2 hours programming)
- mandatory presence (excused absences made up with special homework assignments)

10 homeworks (1 point each)

3 quizzes (20 points each)

2 graded laboratories (60 points each)

- 1-2h individual projects
- main part of final grade

Lecture review questions (1)



What are the six steps of programming?

What part does coding play in program development?

- Program specification
- Program design
- Program coding
- Program testing
- Program documentation
- Program maintenance



ORAL
EXERCISE

Lecture review questions (2)

What is meant by “generation” in reference to programming languages? What is the difference between low-level and high-level languages?

- machine languages (1G) (0100111)
- assembly languages (2G) (MOV A1, A2)
- procedural languages (3G) `for(int i=0; i<n; i++)`
- problem-oriented languages (4G)

`SELECT id FROM objects WHERE field = 1`

- natural languages (5G)

(do they exist yet? Are they just well interfaced 3G and 4G?)



„Filter this audio input for me.“



ORAL
EXERCISE

Lecture review questions (3)



What is the difference between a compiler and an interpreter?

Compiler – translates a higher level program to lower level code, **generates a compiled program** which can be run later on a given machine

Interpreter – translates a higher level **instruction** into machine code and **executes it** before translating further instructions

Note: JAVA both compiles programs, and later interprets them using a „virtual machine” making java machine-independant



ORAL
EXERCISE

Lecture review questions (4)



Only the **CODING** step of software development involves keying statements into a computer.

In the **DESIGN** step a solution is created using programming techniques such as top-down program design, pseudocode, flowcharts, and logic structures.

Program specification is also called **DEFINITION**.



ORAL
EXERCISE

Lecture review questions (5)



Under the rules of top-down design each module should have ONE function(s).

The omission of a semicolon at the end of a statement in Pascal, C++ is an example of a SYNTAX error.

Executing a loop too many times is an example of a LOGIC error.



ORAL
EXERCISE

Algorithm exercises (1)

Determine a step-by-step procedure (list the steps) to do the following tasks:

- Replace a flat tire with the spare tire.
- Make a telephone call.
- Roast a turkey.

Note: There is no single answer for each of these tasks. The exercise is designed to give you practice in converting intuitive commands into equivalent algorithms and understanding the differences in the thought processes involved in the two types of responses.



Algorithm exercises (2)

Write a set of detailed, step-by-step instructions, in English (pseudocode) to calculate the change remaining from a dollar after a purchase is made. Assume that the cost of goods is less than a dollar. The change received should consist of the smallest number of coins possible.



\$0.01



\$0.05



\$0.10



\$0.25



BOARD
EXERCISE

Algorithm exercises (2)

Start with zero QUARTERS, DIMES, NICKELS, PENNIES

CHANGE equals \$1.00 minus PURCHASE

while $\text{CHANGE} \geq \$0.25$ increase QUARTERS by 1
and decrease CHANGE by \$0.25

while $\text{CHANGE} \geq \$0.1$ increase DIMES by 1
and decrease CHANGE by \$0.1

while $\text{CHANGE} \geq \$0.05$ increase NICKELS by 1
and decrease CHANGE by \$0.05

while $\text{CHANGE} \geq \$0.01$ increase PENNIES by 1
and decrease CHANGE by \$0.01

End by giving the customer QUARTERS, DIMES, NICKELS and PENNIES



Algorithm exercises (3)

Propose algorithms to find the smallest number in a group of three integer numbers: a , b , c .

- check all possible combinations („ $a \leq b$ and $a \leq c$ ” ?, „ $b \leq a$ and $b \leq c$ ”... etc.)
- assume a is „smallest”, check if $a \leq b$, if not, b is the new „smallest”, check if „smallest” $\leq c$, if not, c is the new „smallest”



Algorithm exercises (4)

Propose algorithms to sort three integer numbers: a, b, c in ascending order.

- check all possible combinations
(„ $a \geq b \geq c$ ” ? „ $a \geq c \geq b$ ” ... etc.)
- find smallest number, place it at the start, find the smallest from the remaining two, place it as second
- compare neighboring numbers and switch them if they are in wrong order



Top Down Design

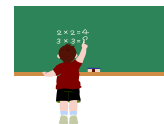
Consider a program specification:

*Input the length and width of a room in metres,
and calculate and display the size of carpet
needed in square metres. [Picking]*

Break the specification down into submodules.

Identify possible variables by looking for nouns
in the program.

Use graphical notation to show the design.



Top Down Design

