

# CIS 15 Midterm Study Guide

This study guide intends to give a brief overview of some of the topics we covered in class. It can be used as a check list, however **you should know everything that we have covered even if it is not mentioned here.**

## Topics

### Unit I

The study sources for this unit: Chapter 1 & 2 from Pohl textbook and lecture notes posted on the course website. It contains topics covered in CIS 1.5 and an introduction to Unix OS shell commands.

- What is compiling, what are the steps involved? Invoking, using g++ compiler.
- Implicit type conversion, what happens when *int a = 3.2;*. how does c++ converts native types?
- Using enumerated data types, (*enum*).
- Random number generation in C++. Using *srand()* and *rand()* functions.
- Declaring and defining functions with or without return types and parameter lists. Calling, using functions.
- Control Structures: *if-else if-else* and *switch-case* statements, *while* and *for* loops.
- What is Unix? Usage of common Unix commands:  
*man, cd, ls, cp, mv, rm, rmdir, mkdir, chmod, pwd*
- Referring to locations using absolute and relative paths in Unix.

### Unit II

Sources: Chapter 4 & 5.1-5.3, 5.7 & 5.8 from Pohl textbook, lecture notes posted on the course website. This unit covers introduction to classes and object oriented programming concepts.

- Definition of *class* and *struct*. Their difference? Declaring and defining *class* and *struct*.
- Creating objects and array of objects of a *class* or a *struct*.
- Accessing member variables and functions with objects using (*.*) operator. public and private access to members.
- Defining and overloading member functions. Both inside class declaration (inline) and outside using scope operator (*::*).
- Keywords, *static* and *const*. What do they mean when used in variable and function contexts?
- Command line arguments. How are they used? :  
*int main(int argc, char \* argv[])*
- What are constructors/destructors? When and how are they defined and used?

- Overloading constructors. Using constructor initializer ( e.g *ctorname(parameter) : membername(parameter){...}* )
- What are copy c'tors? When are they called?
- How does c'tors used as conversion? What does keyword *explicit* mean?
- What are *friend* functions and classes?
- What is composition? What is derivation? The difference between the two?
- Deriving from a class. *public* vs. *private* derivation.

### Unit III

Sources: Chapters 9 & 10 from Pohl textbook, lecture notes posted on the course website. This unit covers C-style strings, C/C++ I/O, file operations and exception handling.

- Difference between C-style and C++ strings. Some common C *string.h* library functions: *strlen()*, *strcpy()* etc.
- C I/O ( *printf()*, *scanf()* ) vs. C++ I/O streams ( *cout*, *cin* ). How are they used?
- C-style character manipulation using *ctype.h* functions: *isdigit()*, *isalpha()*, *isupper()*, *islower()* etc.
- File operations. Opening, reading/writing, closing files. Know the libraries, classes and operators to perform these.
- What is an exception? How can exceptions be handled using *assert()* and *try – throw – catch* blocks.
- How does multiple *catch* works with a single *try*?
- What does keyword *throw* in: *returnType functionName(...)**throw(listofexceptions)* means?