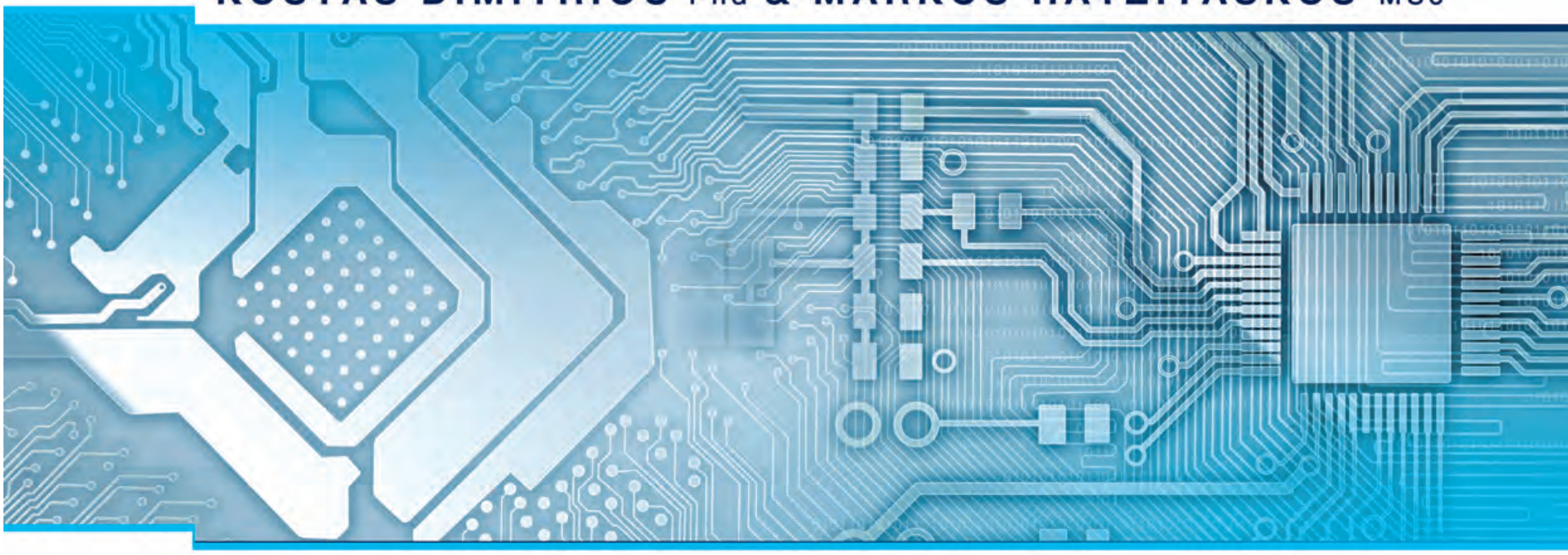
KOSTAS DIMITRIOU Phd & MARKOS HATZITASKOS MSc



CORE COMPUTER SCIENCE

For the IB Diploma Program

(International Baccalaureate)

Express Publishing

Published by Express Publishing

**Liberty House, Greenham Business Park, Newbury,
Berkshire RG19 6HW, United Kingdom
Tel.: (0044) 1635 817 363
Fax: (0044) 1635 817 463
email: inquiries@expresspublishing.co.uk
www.expresspublishing.co.uk**

First published 2015
Third impression 2017

Made in EU

ISBN 978-1-4715-4209-1

Copyright page

List of licensed IB material used:

1. DP Computer Science Guide (first exams 2014):

TOPIC 1- SYSTEM FUNDAMENTALS,1.1 SYSTEMS IN ORGANIZATIONS, Planning and system installation,1.1.1 The context for which a new system is planned,1.1.2 The need for change management,1.1.3 Compatibility issues resulting from situations including legacy systems or business mergers,1.1.4 Different systems implementation,1.1.5 Alternative installation processes,1.1.6 Problems that may arise as a part of data migration,1.1.7 Various types of testing, User focus,1.1.8 Importance of user documentation,1.1.9 Different methods of providing user documentation,1.1.10 Different methods of delivering user training System backup,1.1.11 Causes of data loss.,1.1.12 Consequences of data loss in a specified situation,1.1.13 Range of methods that can be used to prevent data loss Software deployment,1.1.14 Strategies for managing releases and updates,1.2 SYSTEM DESIGN BASICS, Components of a computer system,1.2.1 Hardware, software, peripheral, network, human resources,1.2.1 The roles that a computer can take in a networked world,1.2.3 Social and ethical issues associated with a networked world System design and analysis,1.2.4 Relevant stakeholders when planning a new system,1.2.5 Methods of obtaining requirements from stakeholders.,1.2.6 Appropriate techniques for gathering the information needed to arrive at a workable solution,1.2.7 Suitable representations to illustrate system requirements,1.2.8 Purpose of prototypes to demonstrate the proposed system to the client,1.2.9 Importance of iteration during the design process,1.2.10 Possible consequences of failing to involve the end-user in the design process.,1.2.11 Social and ethical issues associated with the introduction of new IT systems Human interaction with the system,1.2.12 Usability,1.2.13 Usability problems with commonly used digital devices,1.2.14 Methods that can be used to improve the accessibility of systems,1.2.15 Range of usability problems that can occur in a system,1.2.16 Moral, ethical, social, economic and environmental implications of the interaction between humans and machines, TOPIC 2 COMPUTER ORGANIZATION,2.1 COMPUTER ORGANIZATION – COMPUTER ARCHITECTURE,2.1.1 The central processing unit (CPU) and its elements,2.1.2 RAM and ROM,2.1.3 The cache memory,2.1.4 The machine instruction cycle, SECONDARY MEMORY,2.1.5 Persistent storage and secondary memory OPERATING AND APPLICATION SYSTEMS,2.1.6 Functions of an operating system,2.1.7 Software application,2.1.8 Common features of applications, BINARY REPRESENTATION,2.1.9 Bit, byte, binary, decimal and hexadecimal,2.1.10 Data representation, SIMPLE LOGIC GATES,2.1.11 Boolean operators,2.1.12 Truth tables using Boolean Operators,2.1.14 Logic diagrams using logic gates, TOPIC 3. NETWORKS,NETWORK FUNDAMENTALS,3.1.1 Different types of networks,3.1.2 Importance of standards in the construction of networks,3.1.3 Networks, communication and layers,3.1.4 Technologies required to provide a VPN,3.1.5 Use of a VPN,DATA TRANSMISSION,3.1.6 Protocol and data packet,3.1.7 Necessity of protocols,3.1.8 Speed of data transmission across a network.,3.1.9 Compression of data,3.1.10 Characteristics of different transmission media,3.1.11 Packet switching, WIRELESS NETWORKING,3.1.13 Advantages and disadvantages of wireless networks,3.1.14 Hardware and software components of a wireless network,3.1.15 Characteristics of wireless networks.,3.1.16 Different methods of network security,3.1.17 Advantages and disadvantages of each method of network security, TOPIC 4. COMPUTATIONAL THINKING,4.1 GENERAL PRINCIPLES, Thinking procedurally,4.1.1 Procedure appropriate to solving a problem,4.1.2 Order of activities and required outcome,4.1.3 The role of sub-procedures in solving a problem Thinking logically,4.1.4 Decision-making in a specified situation,4.1.5 Decisions required for the solution to a specified problem,4.1.6 Iteration associated with a given decision in a specified problem,4.1.7 Decisions and conditional statements,4.1.8 Logical rules for real-world, Thinking ahead,4.1.9 Inputs and outputs required in a solution,4.1.10 Pre-planning in a suggested problem and solution,4.1.11 Need for pre-conditions,4.1.12 Pre- and post-conditions,4.1.13 Exceptions that need to be considered Thinking concurrently,4.1.14 Parts of a solution that could be implemented concurrently,4.1.15 Concurrent processing and problem solution,4.1.16 Decision to use concurrent processing in solving a problem Thinking abstractly,4.1.17 Examples of abstraction,4.1.18 Abstraction and computational solutions for a specified situation,4.1.19 Abstraction from a specified situation,4.1.20 Real-world and abstraction,4.2 CONNECTING COMPUTATIONAL THINKING AND PROGRAM DESIGN,4.2.1 Searching, sorting and other algorithms on arrays,4.2.2 Standard operations of collections,4.2.3 Algorithm to solve a specific problem,4.2.4 Analyse an algorithm presented as a flow chart,4.2.5 Analyse an algorithm presented as pseudocode,4.2.6 Construct pseudocode to represent an algorithm,4.2.7 Suggest suitable algorithms to solve a specific problem,4.2.8 Deduce the efficiency of an algorithm in the context of its use,4.2.9 Determine number of iterations for given input data,4.3 INTRODUCTION TO PROGRAMMING, Nature of programming languages,4.3.1 State the fundamental operations of a computer,4.3.2 Distinguish between fundamental and compound operations of a computer,4.3.3 Explain the essential features of a computer language,4.3.4 Explain the need for higher level languages,4.3.5 Outline the need for a translation process from a higher level language to machine executable code Use of programming languages,4.3.6 Variable, constant, operator, object,4.3.7 Define various operators,4.3.8 Analyse the use of variables, constants and operators in algorithms,4.3.9 Develop algorithms using loops, branching,4.3.10 Describe the characteristics and applications of a collection,4.3.11 Develop algorithms using the access methods of a collection,4.3.12 Discuss the importance of sub-programmes and collections within programmed solutions.,4.3.13 Construct algorithms using pre-defined sub-programmes, one-dimensional arrays and/or collections, TOPIC 5 OBJECT ORIENTED PROGRAMMING,D.1 OBJECTS AS A PROGRAMMING CONCEPT,D.1.1 The general nature of an object,D.1.2 Distinguishing between object and instantiation,D.1.3 & D.1.4 UML diagrams,D.1.5 Process of decomposition,D.1.6 Relationships between objects,D.1.7 Need to reduce dependencies between objects,D.1.8 Constructing related objects,D.1.9 Data types,D.1.10 Data items passed as parameters,D.2 FEATURES OF OOP,D.2.1 Encapsulation,D.2.2 Inheritance,D.2.3 Polymorphism,D.2.4 Advantages of encapsulation,D.2.5 Advantages of inheritance,D.2.6 Advantages of polymorphism,D.2.7 Advantages of libraries,D.2.8 Disadvantages of OOP,D.2.9 Use of programming teams,D.2.10 Advantages of modularity in program development,D.3 PROGRAM DEVELOPMENT,D.3.1 Class, identifier and variables,D.3.2 Method, accessor, mutator, constructor, signature and return value,D.3.3 Private, protected, public, extends and static,D.3.4 Uses of the primitive data types and the string class,D.3.5 Code examples for D.3.1 - D.3.4,D.3.6 Code example for selection statements,D.3.7 Code examples for repetition statements,D.3.8 Code examples of arrays,D.3.9 Features of programming languages that enable internationalization,D.3.10 Ethical and moral obligations of programmers

2. DP Computer Science guide (first exams 2004) pages 100-102:

```
static void output(String info),{  System.out.println(info);,},static void
output(char info),{  System.out.println(info);,},static void output(byte info),{
System.out.println(info);,},static void output(int info) {
System.out.println(info); },static void output(long
info),{  System.out.println(info);,},static void output(double
info),{  System.out.println(info);,},static void output(boolean info),{
System.out.println(info);,},static String input(String prompt),{  String inputLine
= "";,System.out.print(prompt);,try,{inputLine = (new java.io.BufferedReader( new
java.io.InputStreamReader(System.in))).readLine();},catch (Exception e),{ String
err = e.toString();,System.out.println(err);,inputLine = "";,},return
inputLine;,},static String inputString(String prompt) { return input(prompt);
},static String input(),{ return input("");,},static int inputInt(),{ return
inputInt(""); }, static double inputDouble(), { return inputDouble(""); },static
char inputChar(String prompt),{ char
result=(char)0;,try{result=input(prompt).charAt(0);},catch (Exception e){result =
(char)0;},return result;,},static byte inputByte(String prompt),{ byte
result=0;,try{result=Byte.valueOf(input(prompt).trim()).byteValue();},catch
(Exception e){result = 0;},return result;,},static int inputInt(String prompt),{
int result=0;,try{result=Integer.valueOf(,input(prompt).trim()).intValue();},catch
(Exception e){result = 0;},return result;,},static long inputLong(String prompt),{
long result=0;,try{result=Long.valueOf(input(prompt).trim()).longValue();},catch
(Exception e){result = 0;},return result;,},static double inputDouble(String
prompt) { double result=0; try{result=Double.valueOf(
input(prompt).trim()).doubleValue();},catch (Exception e){result = 0;},return
result;,},static boolean inputBoolean(String prompt) { boolean result=false;
try{result=Boolean.valueOf( input(prompt).trim()).booleanValue();},catch (Exception
e){result = false;},return result;,}
```

KOSTAS DIMITRIOU Phd & MARKOS HATZITASKOS MSc

CORE COMPUTER SCIENCE

For the IB Diploma Program

(International Baccalaureate)

Express Publishing

Kostas’ Dimitriou Dedication

Dedicated to my father who taught me the value of human dignity and to my mother who taught me the value of truth.

Markos’ Hatzitaskos Dedication

Dedicated to my grandmother Elissavet, my mother Eleni-Maria and my brother Kostis, who have always been there for me.

Preface

Kostas Dimitriou holds a PhD in Spatial Decision Support Systems and Environmental Planning, and has taught computer science courses in various undergraduate and postgraduate University courses. He has participated in many scientific conferences and workshops, twenty research projects, and presented sixty scientific articles. He teaches the IB computer science course in the Hellenic American Educational Foundation since 2002. He is a Microsoft Certified Educator**,** Microsoft Expert Educator and Microsoft Expert Education Trainer. {kdimitriou@haef.gr}

Markos Hatzitaskos holds an MSc in Advanced Computing and has taught computer science courses throughout all school levels (from primary to high school and the I.B.). He currently teaches in the Hellenic American Educational Foundation since 2011. In his free time, whenever that might be, he develops mobile applications and attends the Athens School of Fine Arts as an undergraduate. {markosh@haef.gr}

The authors would like to thank the Board of Directors and the Administration of the Hellenic American Educational Foundation (HAEF) for providing an ideal working environment. Thanks are also due to Alkis Dialismas who checked our documents for improper citation or potential plagiarism using Turnitin, and Kostas Ziogas who gave us some valuable advice. Both authors would like to express their gratitude to the employees of Express Publishing and especially to our friend Tzeni Vlachou.

The authors would like to acknowledge the support of John Woodcock, former HAEF IB coordinator, and Sophia Arditzoglou, current HAEF IB coordinator, over the years. A lot of Computer Science students contributed with valuable ideas, comments and suggestions on early drafts. The computer science class of 2014 encouraged us to start this book.

The purpose of this document is to facilitate learning and help our colleagues and CS students around the world. This book is based on the IB computer science syllabus and follows the IB computer science syllabus. The authors did their very best to cite all resources used. If you find a source that is not properly cited please report it to authors. This book was inspired by the book[1]: Jones, R & A. Meyenn. (2004). Computer science Java Enabled. International Baccalaureate. Series, IBID press, Victoria.

[1] Jones, R & Meyenn, A. (2004). Computer science Java enabled. International Baccalaureate. Series, IBID press, Victoria.

The following IBO documents were used during the development of this book:

1. International Baccalaureate Organization. (2004). IBDP Computer Science Guide.
2. International Baccalaureate Organization. (2012). IBDP Computer Science Guide.
3. International Baccalaureate Organization. (2012). IBDP Approved notations for developing pseudocode.
4. International Baccalaureate Organization. (2012). IBDP Java Examination Tool Subset.
5. International Baccalaureate Organization. (2012). IBDP Pseudocode in examinations.

Table of contents