

Undergraduate Degree Learning Outcomes Alignment Template

2012 University of Guelph Undergraduate Degree Learning Outcomes and Associated Skills
reviewed for alignment with Degree Program and/or Specialization Outcomes (BComp:CS)

Critical and Creative Thinking	<i>Critical and creative thinking is a concept in which one applies logical principles, after much inquiry and analysis, to solve problems with a high degree of innovation, divergent thinking and risk taking. Those mastering this outcome shows evidence of integrating knowledge and applying this knowledge across disciplinary boundaries. Depth and breadth of understanding of disciplines is essential to this outcome.</i>	BComp Honours	Computer Science Major
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Inquiry and Analysis	Analyze complex real-world problems. Devise efficient, well-documented computer-based solutions for those problems.	
	2. Problem Solving	Analyze a software development problem. Consider a range of possible approaches to its solution and identify the most promising approaches.	Develop, test, document, deploy, and maintain secure program code to meet given specifications.
	3. Creativity	Apply a knowledge of fundamental algorithms, programming techniques, and design to create software systems.	Design, implement, and evaluate computer-based solutions to meet the needs and constraints of the client.
	4. Depth and Breadth of Understanding	Design, correctly implement and document solutions to significant computational problems. Explain advanced, contemporary concepts related to software development.	Apply knowledge of computing and mathematics to a discipline outside of computing.

Literacy	<i>Literacy is the ability to extract material from a variety of resources, assess the quality and validity of the material, and use it to discover new knowledge. The comfort in using quantitative literacy also exists in this definition, as does using technology effectively and developing visual literacy.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Information Literacy	Apply the core areas of software development. (data structures, theory of computation, operating systems, compilers, programming languages, computer architecture).	Apply knowledge from at least one specialized area of computer science (Human Factors, Security, Graphics, Networks, Databases) to the design/development of software.
	2. Quantitative Literacy	Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems.	Understand the use and structure of the common mechanisms for formally describing software and algorithms. Execute standard measuring and analysis techniques in the evaluation of computing algorithms.
	3. Technological Literacy	Evaluate current techniques, hardware, software, and tools required for the production of software systems. Select the most appropriate tools for the task.	Demonstrate technological fluency. Evaluate and test new technologies systematically.
	4. Visual Literacy	Understand the use and structure of the common mechanisms for formally describing software and hardware structures and graphical user interfaces.	

Global Understanding	<i>Global understanding encompasses the knowledge of cultural similarities and differences, the context (historical, geographical, political and environmental) from which these arise, and how they are manifest in modern society. Global understanding is exercised as civic engagement, intercultural competence and the ability to understand an academic discipline outside of the domestic context.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Global Understanding	Explain how technical constraints limit solutions desired by society in both local and global contexts.	Analyze the local and global impact of computing on individuals, organizations, the environment, and society.
	2. Sense of Historical Development	Summarize the evolution of communication and information technologies, the history of software and the evolution of computer technology.	
	3. Civic Knowledge and Engagement	Analyze the local and global impact of computing on individuals, organizations, and society.	Enumerate common workplace expectations found in information technology (IT) environments.
	4. Intercultural Knowledge and Competence	Recognize the social and cultural responsibilities of a professional working in the discipline of computer science.	

Communicating	<i>Communicating is the ability to interact effectively with a variety of individuals and groups, and convey information successfully in a variety of formats including oral and written communication. Communicating also comprises attentiveness and listening, as well as reading comprehension. It is the ability to communicate and synthesize information, arguments, and analyses accurately and reliably.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Oral Communication	Negotiate, clarify, and document software requirements. Communicate effectively and professionally using audience appropriate technical language.	Communicate confidently and effectively to audiences of a varying technological ability.
	2. Written Communication	Compose written reports to convey technical material meeting accepted standards for writing style.	Produce high quality programming documentation for libraries and reusable code modules.
	3. Reading Comprehension	Locate and evaluate relevant written information related to a software development project, evaluate the contents, and judge the relative importance of the information.	Interpret and use written documentation and programming interface specifications.
	4. Integrative Communication	Communicate effectively with a range of audiences. Present ideas at multiple levels of abstraction and from multiple perspectives.	Interpret, produce, and present work-related documents and information effectively and accurately.

Professional and Ethical Behaviour	<i>Professional and ethical behaviour requires the ability to accomplish the tasks at hand with proficient skills in teamwork and leadership, while remembering ethical reasoning behind all decisions. The ability for organizational and time management skills is essential in bringing together all aspects of managing self and others. Academic integrity is central to mastery in this outcome.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Teamwork	Work effectively as a team, managing conflicts, to accomplish a common goal. Recognize the different roles and responsibilities within a team.	
	2. Ethical Reasoning	Interpret the professional and ethical responsibilities surrounding software development. Demonstrate ethical practises.	Explain the legal and security-related issues arising from design choices for a software system.
	3. Leadership	Generate goals and milestones and recommend task breakdowns for software projects.	Provide technical leadership to software teams.
	4. Personal Organization / Time Management	Recognize the need for, and have the ability to engage in, life-long learning. Consistently organize time and committments to ensure success.	

Undergraduate Degree Learning Outcomes Alignment Template

2012 University of Guelph Undergraduate Degree Learning Outcomes and Associated Skills
reviewed for alignment with Degree Program and/or Specialization Outcomes (BComp:SoftEng)

Critical and Creative Thinking	<i>Critical and creative thinking is a concept in which one applies logical principles, after much inquiry and analysis, to solve problems with a high degree of innovation, divergent thinking and risk taking. Those mastering this outcome shows evidence of integrating knowledge and applying this knowledge across disciplinary boundaries. Depth and breadth of understanding of disciplines is essential to this outcome.</i>	BComp Honours	Software Engineering Major
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Inquiry and Analysis	Analyze complex real-world problems. Devise efficient, well-documented computer-based solutions for those problems.	Use a variety of proven techniques when analysing software development problems.
	2. Problem Solving	Analyze a software development problem. Consider a range of possible approaches to its solution and identify the most promising approaches.	Evaluate possible approaches to solving a problem and explain the benefits and drawbacks to each approach.
	3. Creativity	Apply a knowledge of fundamental algorithms, programming techniques, and design to create software systems.	
	4. Depth and Breadth of Understanding	Design, correctly implement and document solutions to significant computational problems. Explain advanced, contemporary concepts related to software development.	Describe the broad engineering considerations that are the background for developing complex, software-intensive systems.

Literacy	<i>Literacy is the ability to extract material from a variety of resources, assess the quality and validity of the material, and use it to discover new knowledge. The comfort in using quantitative literacy also exists in this definition, as does using technology effectively and developing visual literacy.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Information Literacy	Apply the core areas of software development. (data structures, theory of computation, operating systems, compilers, programming languages, computer architecture).	Translate software specifications into well-documented designs.
	2. Quantitative Literacy	Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems.	Collect, analyse and interpret metrics for software and software development teams.
	3. Technological Literacy	Evaluate current techniques, hardware, software, and tools required for the production of software systems. Select the most appropriate tools for the task.	Plan and execute software engineering processes that effectively use available technology and tools.
	4. Visual Literacy	Understand the use and structure of the common mechanisms for formally describing software and hardware structures and graphical user interfaces.	Produce and interpret a variety of diagrams to represent beginning, middle and final stages of software projects.

Global Understanding	<i>Global understanding encompasses the knowledge of cultural similarities and differences, the context (historical, geographical, political and environmental) from which these arise, and how they are manifest in modern society. Global understanding is exercised as civic engagement, intercultural competence and the ability to understand an academic discipline outside of the domestic context.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Global Understanding	Explain how technical constraints limit solutions desired by society in both local and global contexts.	Identify ways in which advances in software development pose new ethical questions, require new standards, and introduces moral problems and dilemmas.
	2. Sense of Historical Development	Summarize the evolution of communication and information technologies, the history of software and the evolution of computer technology.	Reflect on the appropriateness of different software engineering methodologies for different contexts.
	3. Civic Knowledge and Engagement	Analyze the local and global impact of computing on individuals, organizations, and society.	Describe the additional design considerations required for critical software systems.
	4. Intercultural Knowledge and Competence	Recognize the social and cultural responsibilities of a professional working in the discipline of computer science.	Work effectively on multidisciplinary teams.

Communicating	<i>Communicating is the ability to interact effectively with a variety of individuals and groups, and convey information successfully in a variety of formats including oral and written communication. Communicating also comprises attentiveness and listening, as well as reading comprehension. It is the ability to communicate and synthesize information, arguments, and analyses accurately and reliably.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Oral Communication	Negotiate, clarify, and document software requirements. Communicate effectively and professionally using audience appropriate technical language.	Interact with a project client to elicit input and communicate project progress.
	2. Written Communication	Compose written reports to convey technical material meeting accepted standards for writing style.	Produce written documentation for the design, development, and testing of a complex software project.
	3. Reading Comprehension	Locate and evaluate relevant written information related to a software development project, evaluate the contents, and judge the relative importance of the information.	
	4. Integrative Communication	Communicate effectively with a range of audiences. Present ideas at multiple levels of abstraction and from multiple perspectives.	Create and present software project proposals for clients and funding agencies who may not be software professionals.

Professional and Ethical Behaviour	<i>Professional and ethical behaviour requires the ability to accomplish the tasks at hand with proficient skills in teamwork and leadership, while remembering ethical reasoning behind all decisions. The ability for organizational and time management skills is essential in bringing together all aspects of managing self and others. Academic integrity is central to mastery in this outcome.</i>		
	Learning Outcomes and Associated Skills (2013)	Degree Program Outcomes	Specialization (Major/Minor) Outcomes
	1. Teamwork	Work effectively as a team, managing conflicts, to accomplish a common goal. Recognize the different roles and responsibilities within a team.	Employ group-working skills to provide leadership and mentoring to teammates.
	2. Ethical Reasoning	Interpret the professional and ethical responsibilities surrounding software development. Demonstrate ethical practises.	Make defensible judgements about actions and policies when presented with competing ethical perspectives.
	3. Leadership	Generate goals and milestones and recommend task breakdowns for software projects.	Manage a software project including planning, scheduling, and risk assessment/management.
	4. Personal Organization / Time Management	Recognize the need for, and have the ability to engage in, life-long learning. Consistently organize time and committments to ensure success.	

Critical and Creative Thinking	Critical and creative thinking is a concept in which one applies logical principles, after much inquiry and analysis, to solve problems with a high degree of innovation, divergent thinking and risk taking. Those mastering this outcome show evidence of integrating knowledge and applying this knowledge across disciplinary boundaries. Depth and breadth of understanding of disciplines is essential to this outcome. At the graduate level, originality in the application of knowledge is expected.		
	University of Guelph Learning Outcomes and Associated Skills	B.Comp. CS Honours Degree Program Outcomes	B.Comp. General Degree Program Outcomes
	Independent Inquiry and Analysis	Analyze complex real-world problems. Devise efficient, well-documented computer-based solutions for those problems.	Examine complex real-world problems. Devise efficient, well-documented computer-based solutions for those problems.
	Problem Solving	Analyze a software development problem. Consider a range of possible approaches to its solution and identify the most promising approaches.	Analyze a software development problem. Consider a range of possible approaches to its solution and identify the most promising approaches.
	Creativity	Apply a knowledge of fundamental algorithms, programming techniques, and design to create software systems.	Apply a knowledge of fundamental algorithms, programming techniques, and design to create software systems.
	Depth and Breadth of Understanding	Design, correctly implement and document solutions to significant computational problems. Explain advanced, contemporary concepts related to software development.	Design, correctly implement, and document solutions to significant computational problems.

Literacy	Literacy is the ability to extract information from a variety of resources, assess the quality and validity of the material, and use it to discover new knowledge. The comfort in using quantitative literacy also exists in this definition, as does using technology effectively and developing visual literacy.		
	University of Guelph Learning Outcomes and Associated Skills	B.Comp. CS Honours Degree Program Outcomes	B.Comp. General Degree Program Outcomes
	Information Literacy	Apply the core areas of software development. (data structures, theory of computation, operating systems, compilers, programming languages, computer architecture).	Apply the core areas of software development. (data structures, programming languages, computer architecture).
	Quantitative Literacy	Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems.	Apply mathematical foundations, algorithmic principles, design of computer-based systems.
	Technological Literacy	Evaluate current techniques, hardware, software, and tools required for the production of software systems. Select the most appropriate tools for the task.	Evaluate current techniques, hardware, software, and tools required for the production of software systems.
	Visual Literacy	Understand the use and structure of the common mechanisms for formally describing software and hardware structures and graphical user interfaces.	Understand the use and structure of the common mechanisms for formally describing software and hardware structures and graphical user interfaces.

Global Understanding	Global understanding encompasses the knowledge of cultural similarities and differences, the context (historical, geographical, political and environmental) from which these arise, and how they are manifest in modern society. Global understanding is exercised as civic engagement, intercultural competence and the ability to understand an academic discipline outside of the domestic context.		
	University of Guelph Learning Outcomes and Associated Skills	B.Comp. CS Honours Degree Program Outcomes	B.Comp. General Degree Program Outcomes
	Global Understanding	Explain how technical constraints limit solutions desired by society in both local and global contexts.	Explain how technical constraints limit solutions desired by society in both local and global contexts.
	Sense of Historical Development	Summarize the evolution of communication and information technologies, the history of software and the evolution of computer technology.	Discuss the history of software and the evolution of computer technology.
	Civic Knowledge and Engagement	Analyze the local and global impact of computing on individuals, organizations, and society.	Describe the local and global impact of computing on individuals, organizations, and society.
	Intercultural Competence	Recognize the social and cultural responsibilities of a professional working in the discipline of computer science.	Relate the social and cultural responsibilities of a professional working in the discipline of computer science.

Communication	Communication is the ability to interact effectively with a variety of individuals and groups, and convey information successfully in a variety of formats including oral and written communication. Communication also comprises attentiveness and listening, as well as reading comprehension. It includes the ability to communicate and synthesize information, arguments, and analyses accurately and reliably.		
	University of Guelph Learning Outcomes and Associated Skills	B.Comp. CS Honours Degree Program Outcomes	B.Comp. General Degree Program Outcomes
	Oral Communication	Negotiate, clarify, and document software requirements. Communicate effectively and professionally using audience appropriate technical language.	Explain software requirements using appropriate technical language.
	Written Communication	Compose written reports to convey technical material meeting accepted standards for writing style.	Compose written reports to convey technical material meeting accepted standards for writing style.
	Reading Comprehension	Locate and evaluate relevant written information related to a software development project, evaluate the contents, and judge the relative importance of the information.	Locate and evaluate relevant written information related to a software development project.
	Integrative Communication	Communicate effectively with a range of audiences. Present ideas at multiple levels of abstraction and from multiple perspectives.	Apply appropriate user interface techniques to design systems that are usable by people.

Professional and Ethical Behaviour	Professional and ethical behaviour requires the ability to accomplish the tasks at hand with proficient skills in teamwork and leadership, while remembering ethical reasoning behind all decisions. The ability for organization and time management skills is essential in bringing together all aspects of managing self and others. Academic integrity is central to mastery in this outcome. At the graduate level, intellectual independence is needed for professional and academic development and engagement.		
	University of Guelph Learning Outcomes and Associated Skills	B.Comp. CS Honours Degree Program Outcomes	B.Comp. General Degree Program Outcomes
	Teamwork	Work effectively as a team, managing conflicts, to accomplish a common goal. Recognize the different roles and responsibilities within a team.	Recognize the different roles and responsibilities within a team.
	Ethical Reasoning	Interpret the professional and ethical responsibilities surrounding software development. Demonstrate ethical practises.	Explain the professional and ethical responsibilities surrounding software development.
	Leadership	Generate goals and milestones and recommend task breakdowns for software projects.	Use goals and milestones and recommend task breakdowns for software projects.
	Personal Organization/Time Management	Recognize the need for, and have the ability to engage in, life-long learning. Consistently organize time and commitments to ensure success.	Recognize the need for, and have the ability to engage in, life-long learning. Consistently organize time and commitments to ensure success.