



ADMINISTRATIVE RULES OF MONTANA



10.58.528 COMPUTER SCIENCE

- (1) The program requires that successful candidates:
 - (a) demonstrate knowledge of computer science content, models, important principles, and concepts through:
 - (i) knowledge of, and proficiency in, the use of primitive data types;
 - (ii) understanding of data representation;
 - (iii) knowledge of, and proficiency in, the use of static and dynamic data structures;
 - (iv) knowledge of, and proficiency in, the use of common data abstraction mechanisms (e.g., abstract and generic classes such as stacks, trees, etc.); and
 - (v) effective use, manipulation, and explanation of external data stores – various types (text, images, sound) and various locations (local, server, cloud);
 - (b) demonstrate knowledge of algorithm design, analysis, and implementation in an object-oriented programming language using data structures and abstract data types covering:
 - (i) algorithm problem solving techniques and strategies and design methodologies;
 - (ii) algorithm verification;
 - (iii) algorithm complexity and efficiency; and
 - (iv) knowledge of current programming language and current programming language trends;
 - (c) demonstrate effective design, development, and testing of programs by:
 - (i) using a modern high-level programming language; constructing correctly functioning programs involving simple and structured data types; using compound Boolean expressions; and sequential, conditional, iterative, and recursive control structures;
 - (ii) designing and testing programming solutions to problems in different contexts (i.e., textual, symbolic, numeric, graphic) using advanced data structures;
 - (iii) demonstrating knowledge of and skills regarding the syntax and semantics of two high-level programming languages other than those covered in (b)(iv), their control structures, and their basic data representation;
 - (iv) demonstrating knowledge of and skill regarding program correctness issues and practices (i.e., testing, test data design, and proofs of correctness);

- (v) demonstrating knowledge of and skill regarding at least three different program development environments in widespread use;
 - (vi) demonstrating knowledge of and the ability to construct multi-threaded client-server applications;
 - (vii) demonstrating knowledge of and the ability to construct web sites that utilize complex data bases;
 - (viii) demonstrating knowledge of and the ability to construct artificial intelligence and robotic applications; and
 - (ix) demonstrating knowledge of the principles of usability and human-computer interaction and be able to apply these principles to the design and implementation of human-computer interfaces;
- (d) demonstrate knowledge of computer systems and networks and be able to:
- (i) describe the operation of a computer system, CPU and instruction cycle, peripherals, network components, and applications, indicating their purposes and interactions among them;
 - (ii) demonstrate an understanding of operating systems;
 - (iii) demonstrate an understanding of computer networks; and
 - (iv) demonstrate an understanding of the issues involved in building and fielding mobile services;
- (e) demonstrate an understanding of software engineering and be able to demonstrate an understanding of:
- (i) the difference between computer science and software engineering;
 - (ii) software development methodologies and the software development life cycle; and
 - (iii) the purpose and contents of the software engineering body of knowledge;
- (f) demonstrate an understanding of the key concepts of computer/information security and be able to:
- (i) demonstrate an understanding of the concept of "attack surface" and the various methods used to minimize an attack surface;
 - (ii) demonstrate an understanding of the importance of maintaining logs of all system activity related to security; and
 - (iii) demonstrate an understanding of the purpose and general functionality of a firewall;
- (g) demonstrate an understanding of the role computer science and software engineering plays in the modern world and be able to demonstrate an understanding of:
- (i) significant historical events relative to computers and information systems;
 - (ii) the social, ethical, and legal issues and impacts of computing and information systems;
 - (iii) the contributions that computer and information science and software engineering make to science, the humanities, the arts, commerce, and entertainment;

- (iv) and ability to teach social issues related to the use of computers and information systems in society and the principles for making informed decisions including, but not limited to, security, privacy, intellectual property, equitable access to technology resources, limits of computing, and rapid change; and
- (v) the many different careers that are closely related to the development and use of computer and information systems;
- (h) demonstrate effective content pedagogical strategies that make the discipline comprehensible to students and:
 - (i) design projects that require students to effectively describe computing artifacts and communicate results using multiple forms of media;
 - (ii) identify problematic concepts and constructs in computer science and appropriate strategies to address them; and
 - (iii) promote and model the safe, effective, and ethical use of computer hardware, software, peripherals, and networks and develop digital citizenship.

Authorizing statute(s): 20-2-114, MCA

Implementing statute(s): 20-2-121, MCA

History: NEW, 1991 MAR p. 300, Eff. 3/15/91; AMD, 1992 MAR p. 1475, Eff. 7/17/92; AMD, 1994 MAR p. 2722, Eff. 10/14/94; AMD, 2000 MAR p. 2406, Eff. 9/8/00; AMD, 2007 MAR p. 190, Eff. 2/9/07; AMD, 2014 MAR p. 2936, Eff. 7/1/15; AMD, 2023 MAR p. 86, Eff. 7/1/23.