

Differentiated Instruction Teaching/Learning Examples



GRADE 10 INTRODUCTION TO COMPUTER STUDIES (ICS20): ANIMATION PROGRAMMING—COMPUTER STUDIES

Five 75-minute periods

1. Pair-Share—Viewing Animation (Cooperative Learning)*
2. Guided Viewing/Co-constructing Criteria (Setting Objectives and Providing Feedback)*
3. Jigsaw Review (Cooperative Learning)*
4. Self-assessment and Animation Task Preferences (Setting Objectives and Providing Feedback)*
5. Animation Learning Contract**
6. Peer Assessment and Feedback (Setting Objectives and Providing Feedback)*

*Marzano's Categories of Instructional Strategies (See Resources, below.)

**Differentiated Instruction Structure

DIFFERENTIATED INSTRUCTION DETAILS

Knowledge of Students

Differentiation based on student:

- Readiness Interests Preferences: Styles Intelligences Other (e.g., environment, gender, culture)



Need to Know

- Student readiness (i.e., support required, appropriate level of complexity of animation task), student preferences (i.e., way of working—individually or in pairs), and software preference to design and set up animation task

How to Find Out

- Student completion of Self-assessment and Animation Task Preferences (Appendix B) during the lesson

Differentiated Instruction Response

- Topic, Entry Point (content) Ways of learning (process) Ways of demonstrating learning (product) Learning environment

CURRICULUM CONNECTIONS

Overall Expectations: Introduction to Programming

- B2. Plan and write simple programs using fundamental programming concepts
B3. Apply basic code maintenance techniques when writing programs

Specific Expectations: (B2) Writing Programs; (B3) Code Maintenance

- B2.1 Use a visual problem-solving model to plan the content of a program
B2.2 Use variables, expressions and assignment statements to store and manipulate numbers and text in a program
B2.5 Write programs that use looping structures effectively
B3.1 Write clear and maintainable code using proper programming standards

Learning Goals:

- Apply computer program writing skills and knowledge to produce an animation
- Plan the content of an animation program using a visual problem-solving model

ASSESSMENT AND EVALUATION

Assessment/Success Criteria

Thinking

- Plans the content of a program using a visual problem-solving model

Communication

- Writes code according to standards (e.g., easy to follow, use of tabs and comments to clarify content)

Application

- Uses programming knowledge and skills to create an animation (e.g., smooth transitions, follow-through, timing to convey weight and size, sequence)
- Uses variables, expressions, assignment statements and looping structures

Assessment Tools:

- Rubric
- Rating Scale
- Anecdotal Comments

PRIOR LEARNING

Prior to this lesson, students will have:

- Prior experience using variables, expressions, assignment statements, and looping structures
- Prior experience writing clear and maintainable code using proper programming standards
- Used visual problem-solving models to plan content

MATERIALS AND RESOURCES

Materials:

Pre-selected short computer animations for students to view (Note: The animations should demonstrate a range of complexity from very simple to more complex, be representative of the varying skill levels of the students in the class, and be created with the software available to the class.)

Chart paper/markers

Appendix A1: Jigsaw Note-taking Organizer: Programming Skills and Knowledge—Expert Group A—one per student

Appendix A2: Jigsaw Note-taking Organizer: Problem-Solving Models—Expert Group B—one per student

Appendix A3: Jigsaw Note-taking Organizer: Software Programs—Expert Group C—one per student

Appendix B: Self-assessment and Animation Task Preferences—one per student

Appendix C: Animation Learning Contract—one per student

Appendix D: Animation Rubric—one per student

Appendix E: Animation Checklist—one per student

Internet Resources:

www.alice.org (Alice educational software that teaches students computer programming in a 3D environment)

www.holtsoft.com (Holt Software Associates Inc., for Turing 4.1.1 programming language)

Ministry of Education (2004). *Think Literacy: Cross-Curricular Approaches, Grades 7–12* (Jigsaw, p. 170). See *Think Literacy Library* at www.edu.gov.on.ca/eng/studentsuccess/thinkliteracy/library.html

Resources:

Kagan, Spencer (1994). *Cooperative Learning*. San Clemente, CA: Kagan Publishing.

Marzano, Robert J., Pickering, Debra and Pollock, Jane E. (2001). *Classroom Instruction that Works: Research-Based Strategies for Increasing Student Achievement*. Alexandria, VA: ASCD.

Ministry of Education (2008). *The Ontario Curriculum, Grades 10–12, Computer Studies*.

Teaching/Learning Sequence: Grade 10 Introduction to Computer Studies (ICS20): Animation Programming—Computer Studies

MINDS ON

- Establishing a positive learning environment
- Connecting to prior learning and/or experiences
- Setting the context for learning

Pairs → Pair-Share—Viewing Animation

Students, in pairs, view a few preselected animations (see Materials and Resources on the reverse of this folder) and discuss:

- Which animations appeal to them? Which do not? Why?
- Which animations are simple? Which are more complex? What adds to the complexity of an animation?

Lead a class discussion to determine student animation interests and the various levels of readiness for computer animation (e.g., prior use of various animation software, experience in animation programming, an understanding of the programming required for more complex animations).

Whole Class → Guided Viewing/Co-constructing Criteria

Set the context for learning by explaining to students that they will soon be “contracted” to use their programming skills to work for a client who needs a computer animation. Indicate that they will be able to choose the degree of complexity of the animation based on their prior knowledge and experience and that they will have a choice of software programs. In order to do a good job, they will need to consider the characteristics of an effective animation. See the Animation Checklist (Appendix E) for examples of effective characteristics.

Students:

- As a class, re-examine some of the sample animations viewed in the Minds On Pair-Share activity
- Individually, note characteristics that make the animations effective (e.g., smooth transitions, follow-through, timing to convey weight and size, colour)
- In groups of two or three, share observations and agree upon a list of criteria for an effective animation
- Share their list with the class

Facilitate a discussion to develop a common set of criteria for an effective animation. Post the criteria for reference throughout the lesson.



CONNECTIONS

L: Literacy
ML: Mathematical Literacy
AfL, AoL: Assessment for/of Learning

AfL: Pre-assessment/Pair-Share/
Anecdotal Comments

ACTION

- Introducing new learning or extending/reinforcing prior learning
- Providing opportunities for practice and application of learning (guided → independent)

Remind students that, in order to create animations that meet the above criteria, they will need to plan their work and apply their programming skills and knowledge using software that they are familiar with. Indicate that their animations do not need to be complex in order to be effective.

Students participate in a Jigsaw to review programming knowledge and skills, problem-solving models and software available for their use.

Whole Class/Small Groups → Jigsaw Review

Set up the Jigsaw as follows:

- **Expert Group A—Programming Skills and Knowledge Review**—variables; expressions; assignment statements; loop structures; writing clear and maintainable code to proper programming standards. See Jigsaw Note-taking Organizer: Programming Skills and Knowledge (Appendix A1).
 - **Expert Group B—Problem-Solving Models**—Input Process Output (IPO); Hierarchy Input Process Output (HIPO); chart and diagram; flow chart; storyboard. See Jigsaw Note-taking Organizer: Problem-Solving Models (Appendix A2).
 - **Expert Group C—Software Programs** (e.g., Turing, Alice, C++, Flash Action Script, and Java). Include software that students will be using for the animation programming activity. See Jigsaw Note-taking Organizer: Software Programs (Appendix A3).
- Adjust the Jigsaw topics and provide materials based on the needs of the class and the software available. If the class is large, set up several Expert Groups for each topic or form six Expert Groups by splitting the number of items within each topic. (See Materials and Resources section on the reverse of this folder for more information on Jigsaw structures.)

During the Jigsaw, students:

- Form Home Groups of three; each student receives all the Jigsaw Note-taking Organizers (Appendices A1–A3). After looking over the three topics (Appendices A1–A3), each student selects one of the Expert Groups based on interest
- Work in their chosen Expert Group
- Decide, as an Expert Group, what information to take back to Home Groups and note this on the appropriate Jigsaw Note-taking Organizer (Appendices A1–A3). Consult references if necessary (e.g., textbook, class notes, Internet).
- Return to their Home Group, use their notes to report back to others and take notes on the other two topics using the appropriate Organizers (Appendices A1–A3).

Monitor and support Expert Group work, clarifying as needed, to ensure adequate preparation of notes for Home Groups.

Individuals → Self-Assessment and Animation Task Preferences

Tell students that the Self-assessment and Animation Task Preferences (Appendix B), which they will complete, will be used to help set up and design animation tasks that consider their prior knowledge, experiences, preferences, and personal learning goals.

Students:

- Consider their experience in the Expert Groups and what they learned in their Home Group to complete the Self-assessment and Animation Task Preferences (Appendix B)
- Note one or two things they would like to learn or improve that relate to one or more of the Expert Group topics (e.g., writing clear and maintainable code, gaining experience on a particular software program) as a personal learning goal

Individuals/Pairs → Animation Learning Contract

Based on student responses to the Self-assessment and Animation Task Preferences (Appendix B), set up learning areas and computer workstations (i.e., stations that allow students to work in pairs or as individuals to create a simple animation or a more complex one using their preferred software, while receiving minimal support using a tutorial or help function, or direct teacher support and guidance).

Review the Animation Learning Contract (Appendix C) with students. The Contract requires each student to create a computer animation that involves the use of non-negotiable knowledge and skills; provides options in terms of planning, software program (readiness), and type of animation; and includes a completely optional third section that some students may wish to complete. Assist students as needed in selecting options on the Contract that align with their strengths, experience and personal learning goals.

Review the assessment criteria for an effective animation using the Animation Rubric (Appendix D) and/or the Animation Checklist (Appendix E).



AfL: Observation/Anecdotal
Comments

AfL: Self-assessment/Rating Scale

L: Reading or viewing subject-
specific texts/Learning Contract

AfL: Observation/Anecdotal
Comments

CONSOLIDATION AND CONNECTION

- Helping students demonstrate what they have learned
- Providing opportunities for consolidation and reflection

Individuals/Pairs → Peer Assessment and Feedback

Students:

- Complete their animation based on the choices selected, seeking assistance and feedback as needed
- Using the Animation Checklist (Appendix E), share and discuss their animation with a peer and refine as appropriate
- Submit Learning Contract materials to the teacher for assessment and feedback and note to what extent they have achieved their personal learning goals

AfL: Peer and teacher
assessment/Checklist