

**MVLA**  
**2019-20**  
**COURSE INFORMATION SHEET**

**Course Title:** App and Game Design (BC1018)

**School:** MVHS

**UC/CSU requirement:** G Elective - 1 year required

**Textbook and/or other learning resources:** Shared digital resources on Google Classroom. As we use a variety of purchased digital curriculum components, they will only be accessible to students signed into an MVLA student Google account.

**Student Learning Outcomes:**

This course aims to teach students applied concepts in math, logic, and computer science. Students learn, apply, and demonstrate mastery in core programming topics in instructions, logic, boolean algebra, conditional, loops, functions, classes, algorithms, abstraction, and design. Students will utilize these concepts towards building apps, 2D games, and 3D games for both mobile devices and desktop computers.

Students will learn how to use the C# programming language using the Unity integrated development environment to write, test, and deploy their code. Throughout the stages of software development, student will apply basic to advanced computer science concepts, develop logical thinking, apply problem solving, and implement software management techniques. These programming tools and resources are provided for students to learn and apply programming concepts. Mastery of material learned are demonstrated through the creation applications and games that will implement key concepts.

1. Collaboratively generate ideas, share them and assess their inherent scope and challenges
2. Translate game ideas into playable prototypes.
  - a. Assess prototypes
  - b. Iterate on prototype until they have a viable, playable version.
3. Plan and manage their own development tasks to work effectively within a team.
4. Evaluate games with users using multiple common game user research methods.
5. Manage time wisely, effectively evaluating the impact of potential changes and monitoring deadlines.
6. Develop a literacy and confidence in reading, writing, and modifying computer programs and scripts.

**Assessment and Grading ([BP 5121](#) / [AR 5121](#)):** To ensure that every student has an equal opportunity to demonstrate their learning, the course instructors implement aligned grading practices and common assessments with the same frequency.

1. Grading categories and their percentage weights:

60% - Projects

40% - Classwork/Homework/Daily Log Files/Quizzes

2. Achievement evidence collected within each grading category:

Projects and quizzes are graded using rubrics that are given to students when the project/quiz is assigned.

Classwork is checked off in person during class time and is graded for both completion and meeting the specifications given by the teacher.

Daily log files are graded on completion and quality. Quality is to be measured in two ways - the thoughtfulness and depth of their written reflection (composed of at least four complete sentences), and the quantity and quality of their warm-up/classwork coding assignments.

3. Grading scales:

90-100%	A
80-89.99%	B
70-79.99%	C
60-69.99%	D
50-59.99%	F

4. Homework/outside of class practices ([AR 6154](#))::

Every day students are responsible for creating a daily log file that includes their programming/coding from the day, as well as written notes about what they worked on, where they left off and how it went. These daily log files must include the daily warm-up, any classwork assignments or ongoing project work, and a reflection composed of multiple complete sentences of the student's own writing.

In general, there is little to no homework in this class. When a project deadline is given, students can definitely work on this outside of class time as needed to complete their project by the deadline.

5. Excused absence make up practices ([Education Code 48205\(b\)](#)):

Upon returning from an excused absence, students must notify the teacher and work out a schedule to make up missed work. If the absence is planned ahead of time, the student must notify the teacher before leaving to work out a schedule. In general, a student is given as many days as they were absent to complete the assignment upon their return.

6. Academic integrity violation practices ([MVHS Academic Integrity Policy](#)):

If a student violates the integrity policy (e.g. copy and pastes code from a friend or from the web) on a minor assignment, the student will earn a zero on the assignment, receive a warning, and the parent(s)/guardian(s) will be contacted.

If a student violates the integrity policy on a major assignment (e.g. final semester project), the student will be referred to administration for further disciplinary action.

7. Late work practices:

All work must be completed on time in order to receive full credit.

Unexcused late work can be turned in within one week of the due date and will receive a maximum grade of 90%. If the student is absent on the due date, they have one week from the day that they return to school to complete the missed work. This applies to quizzes and projects.

Daily log files cannot be turned in late, as they are a journal of the work done on that day, as well as how it went and any commentary on the experience.

8. Revision practices:

Most every project and quiz in this class may be revised. Students have one week from the day the assignment grades are posted/returned to complete this revision. Revisions can receive a maximum grade of 90%.

Daily log files cannot be revised, as these are a live record of what was accomplished on a specific day and how it went.

9. Extra credit practices:

There is no extra credit in this course.

10. Additional grading practices:

None

**Instructors' email addresses:**

[brendan.dilloughery@mvla.net](mailto:brendan.dilloughery@mvla.net)

**Additional information:**

This class is about creation. It is about applying your knowledge of math, science, programming with your concept of "what is fun?". With these in mind, as well as the artistic freedom to choose any style, you create applications and games that you want to create.

Recommended Experience - It is strongly recommended, though not required, that a student have experience with computer science before taking this course. This may include taking either "Intro to Computer Science " or "AP Computer Science at Mountain View High School, or it may be experience outside of MVHS. Again, this is only a recommendation for a more enjoyable experience in which the student is better able to apply new ideas of App and Game Design without struggling with the basic concepts of computer science at the same time.