Harvard GSE Creative Computing Curriculum Overview Table

Unit	**Description**	**Key Words, Concepts, Practices**	**Learning Objectives**
0 – Getting Started	Introducing Scratch Scratch Account Design Journal Scratch Surprise Scratch Studio Critique Group	profile editor project page studio critique croup red, yellow, green	Students will: • be introduced to the concept of computational creation, in the context of Scratch • be able to imagine possibilities for their own Scratch-based computational creation • become familiar with resources that support their computational creation • prepare for creating Scratch projects by establishing Scratch accounts, exploring Scratch studios, creating design journals, and organizing critique groups
1 – Exploring	Programmed to Dance Step-By-Step 10 Blocks My Studio Debug It About	experimenting and iterating testing and debugging sequence sprite motion looks sound costume backdrop tips window remix interactive collage pair-share	Students will: • build on initial explorations of the Scratch environment by creating an interactive Scratch project • be introduced to a wider range of Scratch blocks • become familiar with the concept of sequence • practice experimenting and iterating while creating projects
2 - Animations	Performing Scripts Build-A-Band	loops events	Students will:

	Orange Square, Purple Circle It's Alive Debug It Music Video	parallelism control broadcast scripts presentation mode bitmap vector animation gallery walk	• be introduced to the computational thinking concepts of loops, events, and parallelism • become more familiar with the concepts of sequence • experiment with new blocks in the Events, Control, Sound, and Looks categories • explore various arts-themed Scratch programs • create an animated music video project
Unit 3 – Stories	Characters Conversations Scenes Debug It Creature Construction Pass it On	reusing and remixing make a block backpack stage pass-it-on story pair programming scratch screening design demo	Students will: • gain familiarity in and build understandings of the benefits of reusing and remixing while designing • develop greater fluency with computational concepts (events and parallelism) and practices (experimenting and iterating, testing and debugging, reusing and remixing) • explore computational creation within the genre of stories by designing collaborative narratives
Unit 4 – Games	Dream Game List Starter Games Score Extensions Interactions Debug It	abstracting and modularizing conditionals operators data variables and lists sensing feedback fair arcade day puzzle jar brain dump	Students will: • be introduced to the computational concepts of conditionals, operators, and data (variables and lists) • become more familiar with the computational practices of experimenting and iterating, testing and debugging, reusing and remixing, and abstracting and modularizing by building and extending a self-directed maze, pong, or scrolling game project • identify and understand common game mechanics
Unit 5 – Diving Deeper	Know Want Learn Round Two Advanced Concepts	video sensing cloning peer interviews	Learners will: • reflect on past experiences to self-assess current learning goals and needs

	Hardware & Extensions Activity Design My Debug It	hardware extensions	• create a self-remix by extending a previously started project • be introduced to various hardware extensions that connect Scratch to the physical world • gain more fluency in computational concepts and practices by exploring the newest Scratch features (video sensing, cloning) • experiment with designing learning experiences for others
Unit 6 – Hackathon	Project Pitch Project Planning Design Sprint Project feedback Project Check-in Unfocus Group Showcase prep Showcase	hackathon design sprint project pitch unfocus group showcase	Students will: • be introduced to the format of a hackathon event • demonstrate knowledge of computational concepts (sequence, loops, events, parallelism, conditionals, operators, data) and practices (experimenting and iterating, testing and debugging, reusing and remixing, abstracting and modularizing) by defining, developing, and presenting a personally meaningful, self-directed project • have multiple opportunities for collaboration by working in peer teams, sharing skills, and giving and receiving multiple • rounds of feedback