

The Importance of Building Interfaces

Author's Name: Richard Hanson

Coach Name: Ken Amunrud

Host Organization: Agilent Technologies

ETP Type: New Lesson

Subject/Grade: Computer Science/High School



Abstract

The Importance of Building Interfaces is a short (approx 4 days) mini unit that emphasizes why it is valuable and necessary to establish interfaces to be used in larger programming projects. Many students struggle with understanding the rationale behind creating interfaces, and this sequence of lessons will increase student understanding of the concept of an interface and how to best design them.

Focal Standard(s)

21st Century Skills

COMMUNICATION AND COLLABORATION

Collaborate with Others

- Assume shared responsibility for collaborative work, and value the individual contributions made by each team member

Measurable Objective(s)

Students will use effective communication strategies to plan and decide on the interfaces to be used among multiple objects created in this pair of programming projects.

Formative Assessment(s)

Upon assembling the independently programmed parts of the project. Students will discuss the shortcomings and strengths of their design process. Groups will analyze the compile errors in the project and strategize how those errors could have been avoided. Groups will also take the opportunity to have each participant explain their process for solving the portion of code they wrote.

Summative Assessment

Groups will be given a second project to complete that is similar to the first one. Groups will present to the class and the class will have the opportunity to critique the design and process.

21st Century Skills and Applications

In this series of lessons I will assess the 21st Century Skill of Collaboration - The ability to collaborate and effectively plan a project will be developed in this unit and students of all programming abilities will benefit.

Fellowship Description

Agilent Technologies is a science-technology research and development firm. My project finds me working with mechanical engineers helping to write code to provide an interface between a user and a physical device. There is an exciting AR aspect to the project. I am getting a chance to contribute to a project using modern skills and software that are in high demand in the workforce today. In my daily work I work with engineers designing the layout of devices and software specialists who help me develop my programming and workflow skills.

Fellowship Connection to School/Classroom

This fellowship has given me the opportunity to experience the communication that is necessary on projects in the modern workplace. Many of my computer science students will be excited to hear about what they can expect upon entering the workforce. My experiences will enhance my teaching, as well, because it is highlighting some important communication skills and frameworks used in a workplace that is at the leading edge of the STEM industry.

Instructional Plan

Time required: (1-2 50 minute periods, or 1 90 minute block)

Opening.

Students have interacted with a virtual robot with a limited set of skills. The skills the robot comes with include: `move()`, `turnLeft()` and `putBeeper()`. Students have previously used these skills to have robots map out a square and diamond of variable side lengths (determined at runtime by the user).

In this pair of lessons the students will learn the importance of determining an interface and specifications before writing their code. This is a very important skill that is absolutely necessary in any reasonably sized project where many programmers are contributing. In my experience at Agilent Technologies, the first step in developing any class has been to create the interface that the object will be implementing so that other programmers who will use this object know how they are expected to interact with it.

My expectation is that most students will not communicate these specifications properly and their projects will have many issues on this first attempt. On their second opportunity, it is expected that they will spend much more time planning out their specification and interfaces so that the code they come up with as individuals meshes properly and the problem is sufficiently solved. In order for the impact of planning to be most effective, students will not be allowed to interact with each other after their initial 5 minutes of planning time.

Preview: Students will be reminded of their previous experience with robot class, and, if necessary, class will engage in a remediation activity (class writes the code to create another shape, perhaps, with students

taking turns contributing to the program by typing in Java code at the teacher station which is projected for the class).

Group/Individual activity 1: Students are each given 3 letters to map out with a robot that is passed into the function. Students are instructed to create a function that inputs a Robot and instructs that Robot to place beepers in the shape of a letter. This will then be repeated for two more letters (for a total of 3 letters per student). They are given no instruction regarding method naming, initial location or direction of Robot, nor the size of the letters. When students have completed their functions, one student will be asked to fill in a short method that should have the robot spell out a phrase. It is very likely the code will not compile, because it is likely the students did not standardize their naming protocols. If they did manage to plan that out, it is likely their letters will have issues ranging from pointing in the wrong direction, being different sizes, and overlapping other letters. This should provide a great opportunity for class discussion regarding the proper communication that needs to take place for a successful project

Each group/member will be formatively assessed by fixing compile errors as well as communicating with each other what they each thought the specifications for the robot and letters were. There is a template of the zipped project below and linked to [here](#). Students will be able to unzip the file and import the project into Eclipse. After importing, they may work on their letter-writing functions. I have provided pdf versions of each file the students will complete.

Group/Individual activity 2: Numbers and math operation symbols will be assigned to each member of the group (groups can be larger today). Students are instructed that (similar to activity 1) they will be responsible for filling in a method that directs the specified Robot to draw their numbers and/or symbols. Summative assessment will be based on the success of students having their Robot draw a math equation successfully. Grades can be based on success of individual students to complete their task(s) as well as how well the group communicated specifications and established naming conventions before individual coding began. Zipped Eclipse Project is linked to [here](#) and below. Again, I have provided pdf versions of the Java files that students will complete.

Supply List

Kareltherobot.jar (this is the package that has the information about the Robot class)
LetterDrawerABC, LetterDrawerDEF, LetterDrawerMNO, LetterTester
MathSymbol0_4, MathSymbol5_9, MathOperations, MathEquation

Bibliography

[Karel The Robot API](#), [Using karel the robot](#)

Keywords

interface, implementation, standards, standardization, specifications, collaboration

Links to Files in this ETP

Day 1 Files	Day 2 Files
LetterDrawerABC.Java (as Pdf) LetterDrawerDEF.Java (as Pdf) LetterDrawerMNO.Java (as Pdf) SentenceWriter.Java (as Pdf)	EquationMaker.Java (as Pdf) Number0_4.Java (as Pdf) Number5_9.Java (as Pdf) SymbolMaker.Java (as Pdf)
LetterDrawingProject - This is the Zipped Eclipse Project for Day 1 Activity.	EquationDrawingProject - This is the Zipped Eclipse Project for Day 2 Activity.