# Unit 2: Using Objects

Content Area: **Template**
Course(s):
Time Period: **Full Year**
Length: **Full Year**
Status: **Published**

## UNIT RATIONALE

In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to Math class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the String class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course but will be very important in Units 5 and 9 when teaching students how to write their own classes and about inheritance relationships.

## ESSENTIAL QUESTIONS

How can we simulate election results using existing program code?

How are appropriate variables chosen to represent a remote control?

How do the games we play simulate randomness?

## STANDARDS

## NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

### New Jersey (NJSLS) - Grades 9-12 - Computer Science and Design Thinking (2020)

| |
|---|
| **8.1.12.CS.1:** |
| Describe ways in which integrated systems hide underlying implementation details to simplify user experiences. |
| **8.1.12.AP.1:** |
| Design algorithms to solve computational problems using a combination of original and existing algorithms. |
| **8.1.12.AP.2:** |
| Create generalized computational solutions using collections instead of repeatedly using simple variables. |
| **8.1.12.AP.3:** |
| Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice. |

**8.1.12.AP.4:**

Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

**8.1.12.AP.5:**

Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

**8.1.12.AP.6:**

Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

| | |
|---|---|
| CS.9-12.8.1.12.AP.1 | Design algorithms to solve computational problems using a combination of original and existing algorithms. |
| CS.9-12.8.1.12.AP.2 | Create generalized computational solutions using collections instead of repeatedly using simple variables. |
| CS.9-12.8.1.12.AP.3 | Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice. |
| CS.9-12.8.1.12.AP.4 | Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue. |
| CS.9-12.8.1.12.AP.5 | Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. |
| CS.9-12.8.1.12.AP.6 | Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs. |
| CS.9-12.8.1.12.CS.1 | Describe ways in which integrated systems hide underlying implementation details to simplify user experiences. |

## NEW JERSEY STUDENT LEARNING STANDARDS: CAREER READINESS, LIFE LITERACIES AND KEY SKILLS

| | |
|---|---|
| TECH.9.4.12.CT.1 | Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3). |
| TECH.9.4.12.CT.2 | Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a). |

## NEW JERSEY STUDENT LEARNING STANDARDS: COMPUTER SCIENCE AND DESIGN THINKING

See content area standards.

## PRE-ASSESSMENTS

Discussion questions:

Think of an object (this is very vague on purpose). What is the name of your object?

What are some characteristics that all types of this object share?

What are some things that this object can do or that you can do with this object?

# INSTRUCTIONAL PLAN

## MODULE 2

## LESSON 2.8

**Lesson 2.8: String Methods**

In this lesson, students will look at Strings as a sequence of characters and utilize String methods from the java.lang package.

Students will learn about packages, libraries and documentation. The following String methods will be examined in this lesson:

```
name.length()
name.substring(2, 6)
name.indexOf("d")
name.equals("Karel")
name.compareTo("Karel")
```

This lesson corresponds with AP Computer Science A topic 2.7.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Call String methods for a String class<br>• Explain the importance of APIs, documentation and packages in Java |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |

| | |
|---|---|
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the *Bigger Strings?* example.<br>• Explore the *Chopping Strings* example.<br>• Explore the *Object Concatenation* example.<br>• Complete the *Speaking* exercise.<br>• Complete the *toString for Flowers* exercise.<br>• Complete the *Organizing Files* exercise.<br>• Complete the *Concatenating Fractions* exercise.<br>• Complete the *Word Games* exercise.<br>• Complete the *Using Java Documentation* handout. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.7

**Lesson 2.7: String Objects**

In this lesson, students will learn about the immutability of Strings as objects. Once a String object is created, it cannot be changed or manipulated. The only way to change a String value is to reassign the variable with a different String value. Students will also practice String concatenation using operators such as + and += and use escape sequences such as \" and \n. This lesson corresponds with AP Computer Science A topic 2.6.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Create String objects for a String class<br>• Concatenate Strings using operators and escape sequences |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This |

| | |
|---|---|
| | quiz is a quick check for understanding.<br>• Explore the *Immutable Strings* example.<br>• Explore the *String Concatenation* example.<br>• Explore the *Rectangle Dimensions* example.<br>• Explore the *Printing Equations* example.<br>• Complete the *Pretty Printing Operations* exercise.<br>• Complete the *Full Name* exercise.<br>• Complete the *QuoteMachine* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.6

**Lesson 2.6: Calling a Non-void Method**

In this lesson, students will learn how to call a non-void method and return a value from a method by using the keyword return. The return keyword returns a variable or value back to the existing program so it can be used further along in the program. This lesson corresponds with AP Computer Science A topic 2.5.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | Create and call non-void methods with parameters and return values |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the *Rectangle* example.<br>• Explore the *Desks in a Room* example. |

| | |
|---|---|
| | • Explore the *Activity Log* example.<br>• Complete the *Number Games* exercise.<br>• Complete the *Construction Costs* exercise.<br>• Complete the *How Far Away Is…?* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.5

**Lesson 2.5: Calling a Void Method with Parameters**

In this lesson, students will build on what they have learned and discover how to call a void method with parameters. Just as constructors can have parameter values, methods can have formal parameters that affect the state of the object as well. Methods can also be overloaded just like constructors! This lesson corresponds with AP Computer Science A topic 2.4.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | Call non-static void methods with parameters |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the *Rectangle* example.<br>• Explore the *Moving a Point* example.<br>• Complete the *Using the Point Class* exercise.<br>• Explore the *Calculator* example.<br>• Complete the *Basketball Players* exercise.<br>• Complete the *More Operations* exercise. |

| | |
|---|---|
| | • Complete the *Chat Bot 2.0* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.4

**Lesson 2.4: Calling a Void Method**

In this lesson, students will take a deeper look into creating and calling the methods of a class. Methods are procedures that define and allow for control of the behavior of an object. Once an object is instantiated, the instance variables can be used across the different methods that are created. Students will also learn about procedural abstraction for methods that can be used without knowing all of the underlying details and code. This lesson corresponds with AP Computer Science A topic 2.3.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Create and call class methods<br>• Call non-static void methods without parameters |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the *Area of a Rectangle* example.<br>• Explore the *Program Flow* example.<br>• Complete the *Hello!* exercise.<br>• Complete the *Loose Change* exercise.<br>• Complete the *Chat Bot* exercise. |

| | |
|---|---|
| | • Complete the *Greetings and Salutations* challenge activity. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.3

**Lesson 2.3: Overloading**

In this lesson, students are introduced to method overloading. This is when several different methods are written, each with the same name. As long as each method's parameter list is different, the same method name can be used multiple times! This lesson corresponds with AP Computer Science A topic 2.2.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Explain the purpose of method overloading<br>• Create classes that overload methods<br>• Explain how null objects do not point to any particular object data<br>• Create programs that use other classes as a client to solve a specific problem |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the first lesson video and take the corresponding quiz This quiz is a quick check for understanding.<br>• Explore the *Overloaded GrilledCheese* example.<br>• Explore the *Using Scanner as a Client* example.<br>• Explore the *Null Pointer Exception* example.<br>• Complete the *Plain Coffee* exercise.<br>• Complete the *Custom Pinatas* exercise.<br>• Complete the *Website Class* exercise. |

| | |
|---|---|
| | • Complete the *Empty References* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.2

**Lesson 2.2: Creating and Storing Objects (Instantiation)**

In this lesson, students will create and use constructors. The constructor, or signature of a class, allows for the creation of a new object. Students will create objects by calling constructors with parameters. Parameters are values that are passed into a constructor. These are referred to as actual parameters. This lesson corresponds with AP Computer Science A topic 2.2.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Create and use constructors<br>• Create objects by calling constructors with parameters |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the *The Point Class* example.<br>• Explore the *The Student Class* example.<br>• Complete the *Using the Rectangle* exercise.<br>• Complete the *Student GPA Field* exercise.<br>• Complete the *Instance Variables for Your Dog* exercise.<br>• Complete the *Pizza Time!* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.1

**Lesson 2.1: Objects: Instances of Classes**

In this lesson, students are introduced to classes and objects. These are the foundations of object oriented programming. Students will learn about objects that have state and behavior, and classes which are the templates for creating objects. This lesson corresponds with AP Computer Science A topic 2.1.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Explain the relationship between a class and an object<br>• Create classes with instance variables |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities<br>AP Topic Questions |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding<br>AP Topic Questions |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the GrilledCheese Skeleton example.<br>• Explore the Shark Skeleton example.<br>• Complete the Free Response: What instance variables? activity.<br>• Complete the Pizza Instance Variables exercise.<br>• Complete the Phone Skeleton exercise. |
| **Suggested Modifications** | **English Language Learners**<br>**Adjusted Speech**: The teacher changes speech patterns to increase student comprehension. This could include facing the students, paraphrasing, clearly indicating the most important |

ideas, and speaking more slowly.

**Visuals**: The teacher uses graphics, pictures, visuals, and manipulatives. This helps ELL students better understand and comprehend the subjects at hand.

**Front-Loading Vocabulary**: The teacher front loads vocabulary This means providing students with a list of important vocabulary words they will need to know for a book, lesson, etc. prior to the lesson being taught. Including pictures to go with the vocabulary words is also very beneficial for the students.

**Students with Individualized Education Plans/504s**
**Chunking**: The teacher presents information in a way that makes it easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded by excessive detail. The best way to deliver information is to organize it into meaningful units. Because students with special needs get overloaded easily, chunking is an effective strategy to use with them.

**Checking for Understanding**: It is important to constantly chec for understanding, especially for students who have accommodations. Teachers want to make sure students understand the concepts being covered in a way that makes sense to them.

**Extra time**: The teacher provides students with special needs extra time to complete work or answer questions. It is important to give students enough time to process their thoughts.

**Oral Reading**: The teacher will read work orally to students. Class work such as tests and literature circles may need to be read aloud to the student.

**Gifted & Talented Strategies**
**Extensions/Enrichments**: Teachers will provide gifted and talented students with extension/enrichment projects. Students will be challenged to further their understanding, to apply acquired knowledge, and/or to produce something in reference t acquired knowledge.

**Modify/Change Activities**: Teachers will monitor and modify activities to accommodate those students who need to be challenged further. Additional reading, problem-solving, writing,

or project work is necessary for those students who are ready to move on at a rate more accelerated than their peers. In this way G & T students are provided the same opportunity for support as special needs students.

**Students at Risk of School Failure**

**Directions or Instructions**: Make sure directions and/or instructions are given in limited numbers. Give directions/instructions verbally and in simple written format. Ask students to repeat the instructions or directions to ensure understanding occurs. Check back with the student to ensure he/she hasn't forgotten.

**Peer Support:** Peers can help build confidence in other students by assisting in peer learning. Many teachers use the 'ask 3 before me' approach. This is fine, however, a student at risk may have to have a specific student or two to ask. Set this up for the student so he/she knows who to ask for clarification before going to you.

**Alternate or Modified Assignments:** Always ask yourself, "How can I modify this assignment to ensure the students at risk are able to complete it?" Sometimes you'll simplify the task, reduce the length of the assignment or allow for a different mode of delivery. For instance, many students may hand something in, the at-risk student may jot notes and give you the information verbally. Or, it just may be that you will need to assign an alternate assignment.

**Increase One to One Time:** When other students are working, always touch base with your students at risk and find out if they're on track or needing some additional support. A few minutes here and there will go a long way to intervene as the need presents itself.

**Contracts:** It helps to have a working contract between you and your students at risk. This helps prioritize the tasks that need to be done and ensure completion happens. Each day write down what needs to be completed, as the tasks are done, provide a checkmark or happy face. The goal of using contracts is to eventually have the student come to you for completion sign-offs

**Hands On:** As much as possible, think in concrete terms and provide hands-on tasks. This means a child doing math may

| | |
|---|---|
| | require a calculator or counters. The child may need to tape record comprehension activities instead of writing them. A child may have to listen to a story being read instead of reading it him/herself.<br><br>**Tests/Assessments:** Tests can be done orally if need be. Breal tests down in smaller increments by having a portion of the test in the morning, another portion after lunch and the final part the next day.<br><br>**Seating:** Seat students near a helping peer or with quick access to the teacher. Those with hearing or sight issues need to be close to the instruction which often means near the front. |

## LESSON 2.9

**Lesson 2.9: Wrapper Classes: Integers and Doubles**

In this lesson, students will learn how to convert primitive types to object types using a wrapper class. They will learn how Java automatically converts types using autoboxing and unboxing.  This lesson corresponds with AP Computer Science A topic 2.8.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Create Integer and Double objects for wrapper classes<br>• Call Integer and Double methods for wrapper classes |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. Thi quiz is a quick check for understanding. |

| | |
|---|---|
| | • Explore the *Creating Integers* example.<br>• Explore the *Using Doubles* example.<br>• Explore the *Autoboxing Example*.<br>• Complete the *Order Up!* exercise.<br>• Complete the *Currency* exercise.<br>• Complete the *Guess the Number!* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## LESSON 2.10

**Lesson 2.10: Using the Math Class**

In this lesson, students will learn about static methods. **Static methods** are methods that can be called without creating an object.

More specifically, students will discover and practice using static methods in the Java Math class. The following Math class methods will be examined in this lesson:

```
Math.abs(x)
Math.pow(base, exponent)
Math.sqrt(x)
Math.random()
Math.round()
Math.cos()
Math.sin()
Math.PI
```

This lesson corresponds with AP Computer Science A topic 2.9.

| | |
|---|---|
| **Student Learning Intentions (SLI) WALT: (We are learning to…)** | • Call static methods<br>• Evaluate expressions that use the Math class methods |
| **Student Learning Strategies** | Code tracing<br>Create a plan<br>Error analysis<br>Identify a subtask<br>Look for a pattern<br>Pair programming<br>Predict and compare<br>Simplify the problem<br>Think aloud |
| **Success Criteria** | Check for understanding<br>Completion of below activities |
| **Formative Assessment (drives instructional decisions)** | Teacher Observation<br>Check for Understanding |

| | |
|---|---|
| **Activities and Resources** | • Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.<br>• Explore the *Using the Math Class* example.<br>• Explore the *Static Methods: Rectangle* example.<br>• Explore the *Generating Random Numbers* example.<br>• Complete the *Circle Area* exercise.<br>• Complete *The Unit Circle* exercise.<br>• Complete the *Racing* exercise. |
| **Suggested Modifications** | See Lesson 2.1 above. |

## INTERDISCIPLINARY CONNECTIONS: NEW JERSEY STUDENT LEARNING STANDARDS FOR ELA, SOCIAL STUDIES, SCIENCE AND/OR MATHEMATICS

**CCSS - English-Language Arts**

Key Ideas and Details:

CCSS.ELA-LITERACY.RL.11-12.1 Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.

Integration of Knowledge and Ideas:

CCSS.ELA-LITERACY.W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

Production and Distribution of Writing:

CCSS.ELA-LITERACY.W.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Research to Build and Present Knowledge:

CCSS.ELA-LITERACY.W.11-12.7 Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.

Range of Writing:

CCSS.ELA-LITERACY.W.11-12.10 Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences

**CCSS - Mathematics**

Reason quantitatively and use units to solve problems:

CCSS.MATH.CONTENT.HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.

Create equations that describe numbers or relationships:

CCSS.MATH.CONTENT.HSA-CED.A.1 Create equations and inequalities in one variable and use them to solve problems.

Analyze functions using different representations:

CCSS.MATH.CONTENT.HSF-IF.C.7 Graph functions expressed symbolically and show key features of the graph.

Apply geometric concepts in modeling situations:

CCSS.MATH.CONTENT.HSG-MG.A.1 Use geometric shapes, their measures, and their properties to describe objects

Calculate expected values and use them to solve problems:

CCSS.MATH.CONTENT.HSS-MD.A.1 Define a random variable for a quantity of interest by assigning a numerical value to each event in a sample space;

CCSS.MATH.CONTENT.HSS-MD.A.2 Calculate the expected value of a random variable;

**English Language Arts**

- Journal writing
- Close reading of industry-related content
- Create a brochure for a specific industry
- Keep a running word wall of industry vocabulary

**Social Studies**

- Research the history of a given industry/profession
- Research prominent historical individuals in a given industry/profession
- Use historical references to solve problems

**World Language**

- Translate industry-content
- Create a translated index of industry vocabulary
- Generate a translated list of words and phrases related to information technology

**Math**

- Compare and contrast use of equations and variables in algebra and programming.
- Program graphics and use the properties of geometric shapes
- Compare the computer graphic coordinate system with the Cartesian coordinate plane in math
- Compare probability and the use of random numbers in computer programming.
- Track and track various data, such as industry's impact on the GDP, career opportunities or among of individuals currently occupying careers

**Fine & Performing Arts**

- Create a poster recruiting young people to focus their studies on a career in Information Technology

**Science**

- Research the environmental impact of a given career or industry
- Research latest developments in Information technology
- Investigate applicable-careers in STEM fields

## REFLECTIONS