

REVIEW LESSON

MTA Course: Software Development Fundamentals

Lesson name: Software Development Fundamentals 1.1

Topic: Understand computer storage and data types (One 50-minute class period)

File name: SoftDevFund_RL_1.1

Lesson Objective:

1.1: Understand computer storage and data types. *This objective may include but is not limited to:* how a computer stores programs and the instructions in computer memory; memory stacks and heaps; memory size requirements for the various data storage types; numeric data and textual data.

Preparation Details

Prerequisite student experiences and knowledge

Students should have had experience with variables, data types, and displaying text in the console window. This MTA Certification Exam Review lesson is written for students who have learned about programming. Students who do not have the prerequisite knowledge and experiences cited in the objective will find additional learning opportunities using resources such as those listed in the Microsoft® resources and Web links at the end of this review lesson.

Instructor preparation activities

None

Resources, software, and additional files needed for this lesson:

- SoftDevFund_PPT_1.1
- SoftDevFund_SA_1.1
- SoftDevFund_SA_1.1_key

Teaching Guide

Essential Vocabulary:

data type—a definition of a set of data that specifies the possible range of values of the set, the operations that can be performed on the values, and the way in which the values are stored in memory. Defining the data type allows a computer to manipulate the data appropriately. Data types are most often supported in high-level languages and often include types such as real, integer, floating point, character, Boolean, and pointer. How a language handles data typing is one of its major characteristics.

garbage collection—a process for automatic recovery of heap memory.

heap—a portion of memory reserved for a program to use for the temporary storage of data structures whose existence or size cannot be determined until the program is running. To build and use such elements, programming languages such as C and Pascal include functions and procedures for requesting free memory from the heap, accessing it, and freeing it when it is no longer needed. In contrast to stack memory, heap memory blocks are not freed in the reverse order in which they were allocated, so free blocks may be interspersed with blocks that are in use. As the program continues running, the blocks may have to be moved around so that small free blocks can be merged together to form larger ones to meet the program's needs.

memory—a device where information can be stored and retrieved. In the most general sense, the term *memory* can refer to external storage, such as disk drives or tape drives; in common usage, it refers only to a computer's main memory, the fast semiconductor storage (random access memory) directly connected to the processor.

stack—a region of reserved memory in which programs store status data such as procedure and function call addresses, passed parameters, and sometimes local variables.

Lesson Sequence

Activating prior knowledge/lesson staging (5 minutes)

1. Show the Activator slide in the PowerPoint® presentation for this lesson.
 - a. Ask: Identify the components of a computer. (Disk drive, keyboard, mouse, monitor, CPU, motherboard, power supply, memory, etc.)
 - b. Ask: What components are involved in storing program instructions? (RAM, hard drive)

Lesson activity (25 minutes)

1. Show the PowerPoint presentation.
 - a. Explain how a computer stores programs in memory.
 - b. Describe how memory storage involves stacks and heaps.
 - c. Go through the various data types and have students come up with variable declarations using the data types. Give them feedback on the appropriateness of their answers.
 - d. Highlight the numeric data types and focus on their memory size requirements.

Assessment/lesson reflection (20 minutes)

1. Student Activity (SoftDevFund_SA_1.1)

Microsoft Resources and Web Links

Data Types (C# Programming Guide)

<http://msdn.microsoft.com/en-us/library/ms173104%28VS.80%29.aspx>

Suggested best practices:

- If students finish the worksheet early, they should review each other's answers or assist those who need help.

Additional notes to the instructor:

- While the actual range of numbers that each numeric data type can hold is traditionally not emphasized in an introductory course on programming, in this case, it is important, as are the memory size requirements.