eJPT Certification

Section: Introduction to Programming

08/30/2020

Learning Objectives:

* Basic concept of programming
* Typical programming constructs

**Low and High-Level languages:**

* Languages can be split into two groups: high level and low level. Each level tells us how close the language is to the hardware.
  + Low-level languages are interpreted directly by the computer, which means that you can do almost everything in them. However, because of the languages complex nature this can lead to vulnerabilities, if the developer doesn’t have a deep understanding of the language’s capabilities.
    - If you are going to create code that will operate directly on computer memory, then you should use a lower level language.
  + High-level languages typically more user friendly which enables more development but is typically less flexible. If there are no libraries available for the language, then writing custom functions from scratch can be a challenge.
    - Examples of high-level languages are python, java, JavaScript, and visual basic.
    - Compile on the fly using a specific language engine.
    - Programs created using high level languages cannot reside on a bare OS and will need some software already installed on the system to run.
      * Example: java programs require Java Runtime Environment to be installed on your OS.
    - If you want to build a fancy user interface, then you should use a high-level language that has a robust library.

**Programming vs. Scripting:**

* High level programming consists of both programming languages and scripting languages.
* Programming languages require a compiler.
  + What’s a compiler?
    - A compiler converts your source code (what your wrote) into machine readable code (machine code: 0’s and 1’s).
      * Example of languages that require a compiler is C, C++, and Java.
* Scripting languages are usually interpreted, which means that the software environment installed on your computer can read a plain-text program file the same way that you can, and it can execute the instructions without changing the file in anyway.
  + Examples: Perl, PHP, Ruby, Python

**Basic Concepts:**

* All programming and scripting languages share basic concepts.
* Each programming language has its own syntax. Think grammatic rules in human language.
* Syntax various from one language to another and may require specific instructions, such as: ending a statement with a “;” or not.
* **Variables** 
  + Variables can be thought of containers that can be filled with some type of data.
  + Some languages require you to define the variable type. Numbers, words, characters can sometimes be stored in different types of variables.
* **Functions**
  + Functions essentially are blocks of code that can be thought of a mini program inside of a bigger program. Each function is responsible for specific task when called upon.
  + Functions use **arguments** and might **return** a value.
  + Think of a function as a box, and this box needs to take some **argument** (some input), the box then processes this input, and then **returns (**some output) some value.
    - The **arguments** essentially transform inside of the function (the box).
* **Conditional Statements**
  + Every language contains conditional statements
    - A conditional statement means that there is a condition that needs to be checked and there is at least one instruction defined on how to check our condition.
* **Loops**
  + Loops is a form of control flow. It allows to run a particular set of code either an infinite or finite set up times.
    - Loops are often paired with conditional statements in order to check if they should stop, or if they should repeat the instructions again.

Notes

* Programming enables the user to create a set of instructions that a computer can follow.
* Can be used to automate tasks.
* There are several types of programming languages and although they share the same purpose, they all have different syntax and usage requirements.