|  |  |  |  |
| --- | --- | --- | --- |
| Instructor |  | Due Date |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | **1** | **2** | **3** | **4** | Total |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) 1 in your **Python Programming** textbook.

**Part 1 Glossary Terms - Introduction to Computers and Programming Languages**

Define, in detail, each of these glossary terms from the realm of computer programming logic and design and computer topics, in general. If applicable, use examples to support your definitions. Consult your notes or course textbook(s) as references or the Internet by visiting Web sites such as:

|  |  |  |
| --- | --- | --- |
| **http://www.askjeeves.com** | **http://www.webopedia.com** | **http://www.bing.com** |

**(a) ASCII**

|  |
| --- |
| American Standard Code for Information Interchange; ASCII is a standard character encoding for electronic communication. ASCII is a code that maps to binary, octadecimal, hexadecimal, decimal, and HTML forms, which can instruct displays to print characters to a screen, or to execute some sort of action (remove previous character, as when you click “delete” on a keyboard, etc.) |

**(b) Assembly Language**

|  |
| --- |
| An assembly language “assembles” into binary (“machine language”, 1’s and 0’s, which represent High (1) / Low (0) electrical states). It is easier to use assembly commands than to write out binary, however it is notoriously difficult to master. Rollercoaster Tycoon was written entirely in assembly, which is quite an accomplishment… |

**(c) Compiler**

|  |
| --- |
| A program that converts one programming language into another, typically, from a higher programming language into a lower one. |

**(d) High - Level Language**

|  |
| --- |
| Langauges that are meant to be very easy for humans to understand and use, and which compile down to lower level langauges. E.g., JavaScript, Python, Java. |

**(e) Machine Language**

|  |
| --- |
| Binary 0’s and 1’s, or, high level (“on”, or “1”) electrical states (current passing through) and low level (“off”, or “0”) electrical states (current not passing through). Binary systems were invented by Gottfried Leibniz (Newton’s contemporary and rival), and incorporated into his philosophical / religious views but not much else. It wasn’t until Charles Boole published a paper (“Boolean” logic, AND OR NOT) and an MIT student (Shannon, 1937) recognized the similarities between Boolean logic and electrical currents, that binary began to be applied. Then Alan Turing came along and applied these principals to his “Turing machine”, and he and the crew at Bletchley Park expanded on and used these ideas in unlimitedly funded wartime-effort, producing electro-mechanical machines (e.g., the Bomba) that used machine code to operate. From here on, computers were built on binary. Fascinating stuff, really. (sources: “Cryptonomicon” by Neal Stephenson and Wikipedia) |

**Part 2 Exercises - Introduction to Computers and Programming Languages**

Enter T for True or F for False.

**FALSE (1)** Machine languages were developed to make the programmer’s job easier.

**TRUE (2)** A word processor is an example of a systems program.

**TRUE (3)** The compiler guarantees that a program will run correctly.

**TRUE (4)** In object - oriented design, a program is a collection of interacting objects.

**FALSE (5)** The ASCII character set consists of 127 characters. Hint: ASCII is a 7 - bit code

**TRUE (6)** Syntax refers to a set of rules to be followed when writing program.

**FALSE** **(7)** In Python, the necessary steps to execute a program are: edit, interpret, load and execute.

**FALSE (8)** Translating an algorithm into the syntax of Python is the design phase of software development.

**FALSE (9)** The first step to the problem - solving process is to implement the process in a programming language, such as Python, and verify that the process works.

**TRUE (10)** In the structured design approach, a problem is divided into smaller sub - problems, then each sub - problem is solved, and the solutions of all sub - problems are then combined to solve the problem.

**Part 3 Programming Exercises - Introduction to Computers and Programming Languages**

You require a program that calculates discounted prices. You know the original price and the discount percent and you wish to calculate the price after subtracting the discount.

A procedure for the solution of this problem is given below however the steps of the process are not in the proper order. Arrange these steps in the correct order by placing **(1)** next to the first step, **(2)** next to the second step and so on.  
  
Note: there may be more than one solution to accomplish this task

**4** subtract original price by discount amount - save the result as net price

**5** display the net price to the user

**1** input and read the original price

**3** multiply original price by discount percent - save the result as discount amount

**2** input and read the discount percent

**Part 4 Programming Exercises - Introduction to Computers and Programming Languages**

Select the correct answer.

**(1)** Consider the following program.

x = 10

y = 12

z = 2 \* x - 3 \* y

print ("result {0} . format(z))

The missing double - quotes in the code above will be caught by the \_\_**A**\_\_ .

(a) compiler / interpreter (b) assembler

(c) editor (d) control unit

**(2)** To develop a program to solve a problem, you start by \_**A**\_\_\_ .

(a) analyzing the problem (b) implementing the solution in Python

(c) designing the solution (d) entering the solution into a computer system

**(3)** Which of the following is NOT an output device? **D**

(a) monitor (b) CPU

(c) printer (d) secondary storage

**(4)** Which of the following is NOT part of the central processing unit? **D**

(a) control unit (b) instruction register

(c) program counter (d) main memory

**(5)** The first device known to carry out calculations was a(n) \_\_**A**\_\_ .

(a) abacus (b) difference engine

(c) Pascaline (d) None of these