Homework of Introduction to Python Programing…

1. A program must be covered to …… language to be execute by a computer.

   A. assembly b. Machine c. High level d. Very high level

2. is a logical programming language.

   A. PROLOG b. Python c. C# d. Java

3. The program written only using 0's and 1's is

   a. PHP B. High level c. Python d. Machine

4. The founder of Python is

   a. Charles Babbage b. Guido van Rossum c. Dennis Ritchie d. Larry Wall

5. Python is compiled language.

   A. True b. False c. Can't say d. None of these

6. This programing paradigm emerged to remove the reliance on the GOTO statements.

   A. Structured b. Object-oriented c. Logical d. Function

7. Which Python library is popularly referred to as the HTTP library written for humans.

   A. Receive b. Request c. Sockets d. Send

8. In which phase of SDLC does the software developer analyses whether software can be prepared to fulfill all the requirements of the end user?

   A. Design b. Development c. Testing d. Planning

9. This license allows a patent grant for derivative works.

   A. BSD License b. Apache License c. MIT License d. CC License

10. A group of people maintain exclusive control over the source code of software. Such software called.

    A. Freeware b. Shareware c. Proprietary d. Adware

Review questions:

1. What is a programming language?

Answer: Programming Languages: …

Computers cannot write programs on their own as they don't understand human needs unless we communicate with the computer through programming languages. A programming language is a computer language engineered to communicate instructions to a machine. Programs are created through programming languages to control the behavior and output of a machine through accurate algorithms, similar to the human communication process.

2. Briefly explain the steps to install Anaconda.

   Answer:

Step1: Go to the link https://www.continnum.io/downloads. You have the option to download the 32-bit or the 64-bit version of either Python 2.7 or Python 3.6 supported Anaconda distribution. At the time of writing this book, Anaconda supported Python 3.6 version. As and when a new version of Python is released, Anaconda distribution will be updated to newer releases. In this book 64-bit Anaconda distribution supporting Python 3.6 is used to execute programs, so download the same version.

Step2: Click on the executable file of Anaconda Python distribution which you have downloaded and the setup screen will start loading.

Step3: : You will get a welcome screen. Click the next button.

Step4: You will get a License Agreement Screen, read the licensing terms and click on I Agree button.

Step5: : Assuming that you are the only user on your system, select Just Me radio button. Click the next button.

Step6: You need to choose a location to install Anaconda Python distribution. The default installation will be under Users folder. Change the destination folder to C:\Anaconda3 to install Anaconda and click on Next button.

Step7: In the Advanced Installation Options screen, select all the check boxes. Ignore the warnings and click on Install button.

Step8: This starts the installation of Anaconda Python Distribution and once the installation is complete, click on Next button.

Step9: : Finish the setup by clicking on Finish button.

Step10: To check whether the installation is working properly or not, go to the command prompt and type python. You should see a series of lines and a prompt. This is Python Interactive mode. Here, the three greater-than signs “>>>” is the primary prompt of the interactive mode.

3. Describe the steps to install PyCharm.

Answer:

Step1: Go to the link https://www.jetbrains.com/PyCharm/download/. You have the option to download either the PyCharm Professional Edition or the PyCharm Community Edition. If you have purchased a license for Professional Edition then go for it. Otherwise, you can download the Community edition which will suffice most of our requirements. All the programs in this book have been executed using PyCharm Community Edition IDE, so download the same edition.

Step2: Click on the executable file which you have downloaded. You will be presented with the PyCharm Community Edition Setup screen. Click on Next button.

Step3: Now you will be presented with Choose Install Location screen. Go with the default destination folder to install PyCharm Community Edition. Click on Next button.

Step4: Select all the check boxes in the Installation Options screen except for the 32-bit launcher check box (since Windows 10 is a 64-bit OS, you don’t need 32-bit launcher) and click on Next.

Step5: Go with the default Start Menu Folder as shown on the screen and click on Install button. It will take some time for the installation to finish. Once the installation is done click on the Finish button.

Step6: You will be asked whether you want to import previous PyCharm settings. Since we are starting on a clean slate, let's select the second radio button and click on OK button.

Step7: You will be prompted with a Windows Security Alert. Do not worry about it. Click on Allow Access Button. Next screen will be Python community edition initial configuration. Let the default settings remain as it is and click on OK button. After you have completed initial PyCharm configuration, a customization screen will be displayed click on Skip Remaining and Set Defaults button.

Step8: : In the next screen, click on Configure pull down list and select Settings option.

Step9: : In the Default Settings screen, on the left pane, click on Project Interpreter. On the right pane, in the Project Interpreter option, click on the button having toothed wheel icon and select Add. In the Add Python Interpreter screen, on the left pane, click on System Interpreter and select the Python interpreter path from the Interpreter pull down list. Click on OK button.

Step10: It will take some time to list all the packages. Once done click on OK button.

Step11: You will be again presented with the Welcome screen. Now, to work with PyCharm IDE, click on Create New Project option. In the next section, steps to create and execute Python program are discussed in detail.

4. Outline the advantages and disadvantages of machine language.

Answer:

Advantage:

The main advantage of machine language is that it can run and execute very fast as the code will be directly executed by a computer and the programs efficiently utilize memory.

Disadvantages:

- Machine language is almost impossible for humans to use because it consists entirely of numbers.
- Machine language programs are hard to maintain and debug.
- Machine language has no mathematical functions available.
- Memory locations are manipulated directly, requiring the programmer to keep track of every memory location.

5. Why do we need programs? Comment on this.

Answer: We need to program, because Program is set of introduction instructing a computer to do a specific task. A set of instruction called Software.

6. Outline the advantages and disadvantages of High-level language.

Answer:

Advantage:

- Easier to modify, faster to write code and debug as it uses English like statements.

- Portable code, as it is designed to run on multiple machines.

Disadvantage:

The main disadvantage of High-level-language is that, need to compiler or interpreter, because computer can't understand the High-level-language.

7. Give a brief explanation of the history of Python.

Answer: The history of the Python programming language dates back to the late 1980s. Python was conceived in the late 1980s and its implementation was started in December 1989 by Guido van Rossum at CWI in the Netherlands as a successor to the ABC programming language capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community. He is the "Benevolent Dictator For Life" (BDFL), which means he continues to oversee Python development and retains the final say in disputes or arguments arising within the community.

Often people assume that the name Python was written after a snake. Even the logo of Python programming language depicts the picture of two snakes, blue and yellow. But, the story behind the naming is somewhat different.

8. Differentiate between Interpreter and Compiler.

Answer: Compilers: translate source code all at once and the computer then executes the machine language that the compiler produced and Interpreter: An interpreter is a program that reads source code one statement at a time, translates the statement into machine language, executes the machine language statement, then continues with the next statement.

It is the **main** different between Compiler and Interpreter.

9. Mention disadvantages of Assembly language.
    - There are no symbolic names for memory locations.
    - It is difficult to read.
    - Assembly language is machine-dependent making it difficult for portability.
10. Discuss various steps involved in the software development life cycle.

Answer:

Planning of Project: At this stage, the total number of resources required to implement this project is determined by estimating the cost and size of the software product.

Analysis and Requirement Gathering: At this stage, the maximum amount of information is collected from the client about the kind of software product he desires. Different questions are posed to the client like: Who is going to use the product? How will they use the product? What kind of data is given as input to the product? What kind of data is expected as output from the product? Questionnaires enable the development team to gather overall specification of the product in good detail. The software development team then analyzes all these requirements of the client, keeping in mind the design constraints, coding standards and its validity. The aim of analysis and requirements gathering stage is to understand the requirements of the client by all the members of the software development team and see how these requirements can be implemented.

Design: At this stage, the software development team analyzes whether the software can be implemented with all the features as specified by the client. Also, the development team has to convince the client about the financial feasibility and technological viability. The software development team has to select the programming language and the platform to implement the software that is best suited to satisfy the requirements of the client. Software design helps the development team to define and understand the overall architecture required for the software product and the approach is captured in detail in a design document.

Development: At this stage, the development team starts building the software according to the design document. The development team translates the design into a set of programs that adhere to coding standards of their organization. Coding is done by dividing the specification mentioned in the design document into different modules to provide a working and reliable product. This is the longest phase of SDLC.

Testing: At this stage, the software product is tested against the requirements specified by the client to ensure the product is working as expected. The testing team is mainly responsible for checking the system to weed out bugs and to verify that the software product is working as expected. Any bugs that are found in the process or any shortcomings in the features of the software product is conveyed to the software development to rectify. This is the last stage of overall software development before handing over the product to the client.

Deployment: At this stage, the product is released to the client to use after testing the product thoroughly to match the requirements of the client. The client needs to be trained in using the software and documents should be provided containing instructions on how to operate the software in a user-friendly language.

Maintenance: The process of taking care of the developed and deployed software product is known as Maintenance. When the customer starts using the deployed software product, unforeseen problems may come up and these need to be solved. Also, new requirements may come up at the client's workplace and the software needs to be updated and upgraded to accommodate these changes.

11. Give a brief description of open source software.

Answer: The term "Open Source" refers to something people can modify and share because its design is publicly accessible. Open source software is software with source code that anyone can inspect, modify, and enhance.

12. Explain the different types of licenses under which open source software can be released.

Answer:

Apache License:

Apache Software Foundation has authored the Apache License. Apache Software Foundation is well known for various software like Apache HTTP Web Server that powers a large percentage of the Internet's websites, Hadoop that supports the processing of large datasets and many more. There are more than 350 open source initiatives housed at the Apache Software Foundation. Version 2.0 was released in 2004 to make it easier for nonApache projects to use the license and remains one of the most popular licenses to date. Some of the features of this license are,

- It allows you to freely download and use Apache software, in whole or in part, for personal or commercial purposes.

- It requires you to include a copy of the license in any redistribution you may make that includes Apache software.

- Software can be modified to create derivative works and can also be distributed with the same or different license.

- Owner of software cannot be held liable for damages and owner's software trademarks cannot be used in derivative works.

- Significant changes to original software must be noted.

The Apache License allows you to patent the derivative works.

BSD License:

The BSD license is a class of extremely simple and very liberal licenses for computer software that was originally developed at the University of California at Berkeley (UCB) in 1990 and revised twice, being simplified further with each revision. Some of the features of this license are,

- Software can be used commercially.
- Software can be modified to create derivative works which can also be distributed and commercialized.
- Owner of software cannot be held liable for damages and the original copyright and license text must be included with the distributions.

BSD license allows you to commercialize even the derivative works of a software. While Apache license allows a patent grant for derivative works, the BSD license does not allow the same. You are not required to return any modified or improved code to the BSDlicensed software. Due to the extremely minimal restrictions of BSD license, software released under BSD can also be used in proprietary software.

GNU General Public License:

Richard Stallman, an American free software movement activist and programmer launched the GNU Project in September 1983 to create a Unix-like computer operating system composed entirely of free software. With this, he also launched the Free Software Foundation movement and wrote the GNU General Public License (GNU GPL). The GNU GPL is the free software license par excellence. It sets forth distribution conditions that guarantee freedom for its users. A program protected by the GNU GPL is free, but the GNU GPL also stipulates that all programs derived from such a program preserve the same freedom. The license has been revised twice with each revision addressing significant issues that arose in previous versions. Version 3 was published in 2007. Some of the features of this license are,

- Software can be modified to create derivative works and the derivative works can be distributed provided that it is also licensed under GNU GPL 3 as well.
- Distributions must include original software, all source code, original copyright and license text. Owner of the software cannot be held liable for damages.

The GNU GPL guarantees “software freedom” to all users with the freedom to run, copy, distribute, study, modify, and improve the software. GNU GPL license ensures that the open source software remains open source always even when extended into derivative works. However, you should be cautious before incorporating GPL code into your own code as the GNU GPL license requires that you license your entire project under GNU GPL as well. This ensures that the GNU GPL licensed software and its derivatives remain free always.

MIT License:

The MIT license is probably the shortest and broadest of all the popular open source licenses. MIT licensing terms are very loose and more open than most other licenses. Some of the features of this license are,

- Software licensed under MIT license can be used, copied and modified however you want. MIT licensed software can be used with any other project, can be copied any number of times and can be changed however you want.

• MIT licenses software can be given away for free or you can even sell it. There is no restriction on how you intend to distribute the software.

• The only requirement is that the software should be accompanied by the MIT license agreement.

The MIT license is the least protective license out there. It basically says that anyone can do whatever they want with the MIT licensed software, as long as it is accompanied by MIT license.

Creative Commons Licenses:

Creative Commons is a global nonprofit organization that enables sharing and reuse of creativity and knowledge. Creative Commons licenses provide an easy way to manage the copyright terms that attach automatically to all creative material. Creative Commons licenses allow the creative material to be shared and reused. Creative Commons licenses may be applied to any type of work, including educational resources, music, photographs, databases, government and public-sector information, and many other types of material. The only categories of works for which Creative Commons does not recommend its licenses are computer software and hardware. The latest version of the Creative Commons licenses is version 4.0. A Creative Commons license has four basic parts, which can be endorsed individually or in combination. You will find a brief description of each part below.

Attribution: All Creative Commons licenses require that others who use your work in any way must give you credit the way you request, but not in a way that suggests you endorse them or their use. If they want to use your work without giving you credit or for endorsement purposes, they must get your permission first. Beyond that, the work can be modified, distributed, copied and otherwise used.

Share Alike: You let others copy, distribute, display, perform, and modify your work, as long as they distribute any modified work on the same terms.

Non-Commercial: You let others copy, distribute, display, perform and modify and use your work for any purpose other than commercially unless they get your permission first.

No Derivative: You let others copy, distribute, display and perform only original copies of your work. If they want to modify your work, they must get your permission first.

As mentioned before, these parts of the Creative Commons licenses can be mixed together. The least protective Creative Commons license would be the “Attribution” license, which means that as long as people credit you, they can do whatever they like with the work. The most protective Creative Commons license would be the “Attribution, Non-Commercial, No Derivatives” license, which means that others can freely share your work, but should not change it or charge for it, and they must attribute it to you as the creator of the work. Creative Commons licenses give everyone from individual creators to large companies and institutions a simple, standardized way to grant copyright permissions to their creative work.

Present by: **Habibullah Madadi** from ISE class, 1st semester.