**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **05-06-2020** | **Name:** | **Dhanya Shetty** |
| **Course:** | **DIGITAL DESIGN USING HDL** | **USN:** | **4AL17EC026** |
| **Topic:** | **1.VERILOG TUTORIALS AND PRACTICE PROGRAMS**  **2.BUILDING/DEMO PROJECTS USING FPGA** | **Semester & Section:** | **6th A** |
| **Github Repository:** | **Dhanya Shetty\_026** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| C:\Users\Hp\Desktop\report\Screenshot_20200605-153616.png  **C:\Users\Hp\Desktop\report\Screenshot_20200605-153634.png**  **C:\Users\Hp\Desktop\report\Screenshot_20200605-153711.png**  C:\Users\Hp\Desktop\report\20200605_193447.jpg  C:\Users\Hp\Desktop\report\20200605_193457.jpg |
| |  |  |  | | --- | --- | --- | | **Date: 05June2020** |  | **Name: Dhanya Shetty** | | **Course: Python** |  | **USN:4AL17EC026** | | **Topic: section 28**  **Webscraping with Python Beautiful Soup** |  | **Semester & Section:6th A** | |
|  |

|  |
| --- |
|  |
|  |
|  |
| **AFTERNOON SESSION DETAILS** | |
| Image of sessionsC:\Users\Hp\Desktop\report\5JUNE1111111.PNG  C:\Users\Hp\Desktop\report\5JUNE22222.PNG  C:\Users\Hp\Desktop\report\5JUNEPY3333.PNG  The any() function  >>> lines = ["trees are good", "pool is fresh", "face is round"]  >>> website\_list = ["face", "clock", "trend"]  >>> for line in lines:  ... any(website in line for website in website\_list)  ...  False  False  True  We start iterating over the items of website\_list using a for loop. In the first iteration we would have:  any(website in "trees are good" for website in website\_list)  Inside the parenthesis of any() there's another loop that iterates over website\_list:  ("face" in "trees are good")  "clock" in "trees are good")  ("trend" in "trees are good")  If any of the above is True you get the expression evaluated to True. In this case none of them is True, so you get False.  If you want to return True (if all of them are True), use all() instead of any().  So, the part any(website in line for website in website\_list) will either be equal to True or False.  OOP – Object Oriented Programming  *Object-oriented programming* (OOP) refers to a type of computer programming (software design) in which [programmers](https://www.webopedia.com/TERM/P/programmer.html) define the [data type](https://www.webopedia.com/TERM/D/data_type.html) of a [data structure](https://www.webopedia.com/TERM/D/data_structure.html), and also the types of operations ([functions](https://www.webopedia.com/TERM/F/function.html)) that can be applied to the data structure.  In this way, the data structure becomes an [object](https://www.webopedia.com/TERM/O/object.html) that includes both [data](https://www.webopedia.com/TERM/D/data.html) and functions. In addition, programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects.  *The Basic OOP Concepts*  If you are new to object-oriented programming languages, you will need to know a few basics before you can get started with code. The following Webopedia definitions will help you better understand object-oriented programming:  [The Challenges of Cloud Integration](https://o1.qnsr.com/cgi/r?;n=203;c=1639922;s=9534;x=7936;f=201705191426590;u=j;z=TIMESTAMP;k=https://assetform.webopedia.com/controller?asset=175119710&srvid=95900&vkey=4190310&io=11111&qset=CONTACTFORM_HQB&formHQB=y&domain=www.webopedia.com&widgetsrc=mwidget)   * [Abstraction](https://www.webopedia.com/TERM/A/abstraction.html): The process of picking out (abstracting) common features of objects and procedures. * [Class](https://www.webopedia.com/TERM/C/class.html): A category of objects. The class defines all the common properties of the different objects that belong to it. * [Encapsulation](https://www.webopedia.com/TERM/E/encapsulation.html): The process of combining elements to create a new entity. A procedure is a type of encapsulation because it combines a series of computer instructions. * [Information hiding](https://www.webopedia.com/TERM/I/information_hiding.html): The process of hiding details of an object or function. Information hiding is a powerful programming technique because it reduces complexity. * [Inheritance](https://www.webopedia.com/TERM/I/inheritance.html): a feature that represents the "is a" relationship between different classes. * [Interface](https://www.webopedia.com/TERM/I/interface.html): the languages and codes that the applications use to communicate with each other and with the hardware. * [Messaging](https://www.webopedia.com/TERM/M/message_passing.html): Message passing is a form of communication used in parallel programming and object-oriented programming. * [Object](https://www.webopedia.com/TERM/O/object.html): a self-contained entity that consists of both data and procedures to manipulate the data. * [Polymorphism](https://www.webopedia.com/TERM/P/polymorphism.html): A programming language's ability to process objects differently depending on their data type or class. * [Procedure](https://www.webopedia.com/TERM/R/routine.html): a section of a program that performs a specific task.   *Advantages of Object Oriented Programming*  One of the principal advantages of object-oriented programming techniques over procedural programming techniques is that they enable programmers to create [modules](https://www.webopedia.com/TERM/M/module.html) that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its [features](https://www.webopedia.com/TERM/F/feature.html) from existing objects. This makes object-oriented programs easier to modify.  *OOPL - Object Oriented Programming Languages*  An *object-oriented programming language* (OOPL) is a high-level [programming language](https://www.webopedia.com/TERM/P/programming_language.html) based on the object-oriented model. To perform object-oriented programming, one needs an object-oriented programming language.  Many modern programming languages are object-oriented, however some older programming languages, such as [Pascal](https://www.webopedia.com/TERM/P/Pascal.html), do offer object-oriented versions. Examples of object-oriented programming languages include [Java](https://www.webopedia.com/TERM/J/Java.html), [C++](https://www.webopedia.com/TERM/C/C_plus_plus.html) and [Smalltalk](https://www.webopedia.com/TERM/S/Smalltalk.html).  *The First OOPL*  Simula, developed in the 1960s at the Norwegian Computing Center in Oslo, is considered to be the first object-oriented programming language. Despite being first, [Smaslltalk](https://www.webopedia.com/TERM/S/Smalltalk.html) is considered to be the only true object-oriented programming environment and the one against which all others must be compared. It was first developed for educational use at Xerox Corporation's Palo Alto Research Center in the late 1960s and released in 1972. | |
|  | |