|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Key Concepts** | **Explore concepts' significance and relevance** | **Establish relevance, make sense and meaning -Find real-life contexts** | **Establish relevance, make sense and meaning -Find interdisciplinary connections** | **Engage in critical thinking** | **Technology, tools and techniques** | **Plan**  **project management** | **Project specification and sketch** |
| Programming Testing is imperative as it might bring about mission disappointment, effects on operational execution and dependability if not done legitimately.  *Who should test?*  -Developer  -Independent Tester  The Testing Team.  -Program Manager  -QA Lead  -Test Analyst/Lead  -Test Engineer  Once the coding is done, the expert experiences it and tests the framework for the item prerequisites. | Programming testing is truly required to bring up the imperfections and blunders that were made amid the improvement stages.  It's fundamental since it ensures the Customer's unwavering quality and their fulfillment in the application.  It is critical to guarantee the Quality of the product.  Quality item conveyed to the clients helps in picking up their certainty. (Know more about Software Quality).  Testing is required for a compelling execution of programming application or item. | A program that functions admirably for one individual in a given circumstance may demonstrate insufficient or improper for someone else or circumstance. For instance, a word processor with numerical images and an arrangement of apparatuses for situating and controlling them may be perfect for a school teacher composing a material science course reading yet unwieldy and irritating for an author. On the other hand, a straightforward content tool might be favored by the author yet be dismisses by the teacher. Setting driven testing spins around the way that there is no single "best arrangement" that applies to all cases. | Various supporting interdisciplinary connection are:  -Configuration management  -Infrastructure as code  -Documentation  -Software Quality assurance  -User Experience | At the point when directing trials, executing test automation scripts and checking for bugs in the code, quality confirmation individuals may feel a smidgen like they're on auto-pilot. Be that as it may, to go past essentially deciding the level of usefulness offered by an application and guarantee that it is of the most noteworthy quality conceivable, programming analyzers should influence their basic intuition abilities. | Mostly, we are using python to meet all our coding needs. One of the most popular testing tool for python files is **Nose,** which we are going to perform various test cases on different parts of our code. Another popular testing tool is **py.test.**  The py.test structure sets another standard for Python testing, and is prevalent with numerous engineers today. The exquisite and Pythonic colloquialisms it presented for test composing have made it workable for test suites to be composed in a significantly more reduced style | Currently, we are learning about various testing tools and technologies. We are also learning about *Blackbox* and *Whitebox* testing. Also, we will write some intuitive tests to test almost every part of our code.  Immediate deliverables are nothing, but in future a document on testing will be delivered.  **PISE-PBL Sub-Project 6: SE Sub-Subject 6 from October 7-10, 2015** | As mentioned above.  ---------do-------- |