**TESTING**

**CLASS EXERCISE 1**

**SOLUTION**

**QUESTION 1 *(TESTING / VALIDATION)***

* **IDENTITY NUMBER**

Only numerics. Length must be 13.

Can calculate and compare Check Digit. Can make use of an API to validate other digits in the ID number.

* **GENDER**

Only allow M or F to be entered.

* **SURNAME**

Only allow letters & special character to be entered. No numbers.

* **FIRST NAME**

Only allow letters & special character to be entered. No numbers.

* **CELL NUMBER**

Only allow numbers to be entered.

Ensure that not too long (10 digits only).

* **EMAIL**

Check format of email. (Has @ and 1 or 2 .)

* **START DATE**

Ensure date entered in correct format. Must be a valid date (days in month, leap year).

* **END DATE**

Ensure date entered in correct format. Must be a valid date (days in month, leap year).

Check that END DATE is not before / earlier than START DATE.

**QUESTION 2 *(TERMS & DEFINITIONS)***

***Match the TERMS and DEFINITIONS.***

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Stress testing | **J** | 1. Component integration testing | **H** |
| 1. Quality assurance | **M** | 1. Baseline | **V** |
| 1. Accuracy | **E** | 1. Alpha testing | **F** |
| 1. Beta testing | **G** | 1. Unit testing | **A** |
| 1. Big bang testing | **I** | 1. Coverage testing | **O** |
| 1. System testing | **B** | 1. Black box testing | **L** |
| 1. Usability testing | **N** | 1. Acceptance testing | **C** |
| 1. Desk checking | **R** | 1. Quality | **T** |
| 1. Decision table | **Q** | 1. Regression testing | **K** |
| 1. Exception handling | **U** | 1. White box testing | **S** |
| 1. Bottom up testing | **P** | 1. Defect | **D** |

1. Done by developers to ensure their code is working fine & meets the user specifications.
2. Testing the compatibility of the application with the system.
3. Done to ensure that the requirements of the specifications are met.
4. A flaw in a component or system that can cause the component or system to fail to perform its required function.
5. The capability of the software product to provide the correct or agreed results or effects with the needed degree of precision.
6. Testing done at the developers’ site. It is done at the end of the development process.
7. Done at the customers’ site. It is done just before the launch of the product.
8. Testing a collection of items to validate that they work together.
9. A type of integration testing in which software elements, hardware elements, or both, are combined all at once into a component or an overall system, rather than in stages.
10. Gain insight into how the application will perform in unusual situations (e.g. high volume of transactions, concurrent users, etc).
11. Running all your previous test cases against the new version of your application.
12. To ensure a system can do what it should do without being concerned about how it does it.
13. Reviewing and auditing project deliverables to verify if they comply with the applicable standards & guidelines adopted by your organization.
14. Testing to determine the extent in which the software product is understood, easy to learn and easy to operate.
15. Testing of all code paths. Ensures that all lines of code are tested.
16. An incremental approach to integration testing where the lowest level components are tested first and then used to facilitate the testing of higher level components.
17. Shows combination of inputs and/or causes with their associated outputs and/or actions (effects) which can be used to design test cases.
18. Testing of software or a specification by manual simulation of its code.
19. Looks at code and then creates test cases that exercise it.
20. Degree to which a component, system or process meets the specified requirements.
21. Behaviour of a component or system in response to erroneous input, from either a human user or from another component or system, or to an internal failure.
22. A specification or software product that has been formally reviewed or agreed upon, that thereafter serves as the basis for further development and that can be changed only through a formal change control process.