Name: Jackson Debbarma

Roll number: AUFS007170

**Q1.**

**Q**. What is ACID properties and BASE properties?

**Ans**:

|  |  |
| --- | --- |
| **ACID** | **BASE** |
| It stands for “Atomicity Consistency Isolation Durability” | It stands for “Basically Available Soft State Eventual Consistency” |
| Steps of transactions are all complete or none really take place. | In the case of BASE, upon systems request it will response irrespective the data is available or not |
| It preserves the whole operation process and is never a half baked, which is ideal. | System changes are constant, data overriding possibility by recent data. |
| Only valid data are saved | May obstruct consistent data update, for a period of time, until it reaches its consistent level |
| It isolates a transaction, which means any transaction can happen simultaneously, without interference between the process. | It is suitable for the industry where timely data consistency is not always required and is very volatile in nature, for example, the Twitter feed |
| In the case of system failure, data won’t be corrupted and is saved with integrity. |  |

**Q.** What are the differences between paging and segmentation?

**Ans:**

|  |  |
| --- | --- |
| **PAGING** | **SEGMENTATION** |
| It is Transparent to the programmer (system allocates memory) | It involves the programmer (allocates memory to specific function inside code) |
| No shared code | Share code |
| The address generated by CPU is divided into Page number (p) – used as an index into a page table that contains the base address of each page in physical memory.  No separate protection | The logical address consists of two tuples: Segment-number and Offset  Separate protection |
| Computer memory is divided into small partitions that are all the same size and referred to as, page frames. Then when a process is loaded it gets divided into pages that are the same size as those previous frames. The process pages are then loaded into the frames. | Computer memory is allocated in various sizes (segments) depending on the need for address space by the process. |

**Q.** Explain the OOPS principles?

**Ans**: The four principles of object-oriented programming are encapsulation, abstraction, inheritance, and polymorphism.

**Polymorphism** gives a way to use a class exactly like its parent so there’s no confusion with mixing types. But each child class keeps its own methods as they are.

This typically happens by defining a (parent) interface to be reused. It outlines a bunch of common methods. Then, each child class implements its own version of these methods**.** A prominent example could be our cursor changing shape in real-time.

**Encapsulation**: Encapsulation is achieved when each object keeps its state private, inside a class. Other objects don’t have direct access to this state. Instead, they can only call a list of public functions — called methods.

So, the object manages its own state via methods — and no other class can touch it unless explicitly allowed. If you want to communicate with the object, you should use the methods provided. But (by default), you can’t change the state.

### **Abstraction** can be thought of as a natural extension of encapsulation. In object-oriented design, programs are often extremely large. And separate objects communicate with each other a lot. So maintaining a large codebase like this for years — with changes along the way — is difficult. Abstraction is a concept aiming to ease this problem. Applying abstraction means that each object should only expose a high-level mechanism for using it.

This mechanism should hide the internal implementation details. It should only reveal operations relevant to the other objects.

### 

### **Inheritance**

Inheritance helps us to reuse the common logic and extract the unique logic into a separate class. It means that you create a (child) class by deriving from another (parent) class. This way, we form a hierarchy. The child class reuses all fields and methods of the parent class (common part) and can implement its own (unique part).

**Q.** What will happen when you write www.google.com in your web browser.

**Ans:**

1. An IP address of www.google.com is to be checked in the cache for the DNS record by our browser.
2. If the requested URL is not in the cache, ISP’s DNS server initiates a DNS query to find the IP address of the server that host www.google.com
3. The TCP connection is initiated by the browser
4. An HTTP request is sent to the webserver
5. The server handles the request and sends back a response
6. The server sends out an HTTP response
7. HTTP content of google is displayed in our computer’s browser

**Q.2**

**Ans:** (A)

**Q3**

**Ans:** (A)

**Q4**

**Ans:** C

**Q5**

**Ans**: (B)