FAQ

Module-2	
Question 1.	**What's the primary difference between classes and objects in Java?**
Answer	Classes in the Java programming language serve as fundamental blueprints ortemplates that facilitate the creation of objects. Objects are exemplifications of classes, which symbolise things found in the physical world. These entitiespossess attributes that define their current conditions and methods that describe their actions or functionalities.
Question 2.	**How does inheritance promote code reusability in Java?**
Answer	The concept of inheritance facilitates the process by which a newly createdclass, referred to as a subclass or derived class, may acquire the properties and functions of an already existing class, known as a superclass or parent class. This implies that there is no need to rewrite common functionality sincethey are automatically inherited, hence facilitating the reuse of code.

Question 3.	**Could you perhaps provide an explanation of the primary distinctionbetween abstraction and encapsulation?**
Answer	The concept of abstraction serves to conceal intricate details while emphasising fundamental elements, often achieved via the use of abstractclasses and interfaces. In contrast, encapsulation refers to the process of combining data (attributes) and the corresponding methods that manipulatethe data into a cohesive entity. This practice guarantees the security of the data by preventing unauthorised access and change.
Question 4.	**What is polymorphism, and how does it benefit Java programming?**
Answer	Polymorphism, a concept denoting "multiple forms," facilitates the treatmentof items belonging to distinct classes as objects of a shared superclass. The main advantage is in its versatility, as it allows for the referencing of objects from many classes using a single interface. At runtime, the appropriate function is invoked depending on the specific type of the object.

Question 5.	**What are the reasons for a developer to use interfaces in the presence ofabstract classes?**
Answer	Both interfaces and abstract classes provide a means of abstraction. However,interfaces are particularly advantageous in situations when a class requires inheritance of behaviours from various origins, since Java does not enable multiple inheritance with classes. Interfaces serve as a means to establish contractual structures, whereby the implementing class is obligated to definethe designated methods.