K214553_Q1

1. Which Feature of OOP illustrated the code reusability? Explain with examples.

ANSWER:

Inheritance is the feature of OOP that perfectly illustrates the code reusability. Inheritance means inheriting features. The real-world example of inheritance is parents and children. It is observed that children inherit features of their parents. These features might include height, complexion, color of hair and eyes, voice, language and their way of talking etcetera. Similarly, in Object oriented programming there can be a parent class “DRAW” which includes basic data members such as height, base, radius, sides etcetera, and then these data members can be inherited in child class “drawRectangle”. “drawRectangle” will use “sides” data members from parent class DRAW and it will have its own methods such as input() print() drawshape() etc. Likewise, more child classes can be made by using parent class to draw different shapes.
Thus, by using DRAW class as a template we can see code reusability which results in shortening the length of our code and increases efficiency. In short, Inheritance provides code reusability which means we can add more features to an existing class without modifying it.

2. What is the difference between encapsulation and Abstraction? Explain with a real time example.

ANSWER:

Encapsulation literally means bringing data together like in a capsule. In this feature of OOP, we wrap attributes and methods into a single unit. Encapsulation refers to data hiding from real world, at **implementation level**. Logging into an email account can be an example for encapsulation. Composing and sending mail is our concern and we are not concerned how email is verified and what coding is behind all of this sending and receiving process.

Whereas, abstraction refers to data hiding at designing level. It hides unwanted details and gives **relevant details** to the real world for **specific object**. Abstraction focuses on what the object works and hiding how it does it. For example, while writing a code some data members are private and are inaccessible in real world thus abstraction is applied. Abstraction can also be seen when we call a function using an object and we do not see whole lot of method defined in it and we just implement our function easily. Outlook of your mobile phone can be taken as an example of abstraction, for when we make a call only the call screen is visible to us and rest of the screens remains hidden at the moment.