

In the scenario described, inheritance can be used to model the relationships between the various courses and stages of study offered by the university.

For example, we could define a base class called **Course** that contains properties such as the course name, course code, and course duration. We could then define derived classes such as **ICTCourse**, **LawCourse**, and **BusinessCourse** that inherit from the **Course** base class and add their own properties and methods specific to their respective fields of study.

Similarly, we could define a base class called **DegreeCourse** that contains properties such as the degree name and degree level. We could then define derived classes such as **BscITCourse** that inherit from the **DegreeCourse** base class and add their own properties and methods specific to the BSC-IT course, such as the three stages of study.

Using inheritance in this way allows us to model the relationships between the various courses and stages of study in a hierarchical and modular fashion. We can then define additional derived classes as needed to capture more detailed information about the students and their progress through the various courses and stages of study.

For example, we could define a class called **Student** that contains properties such as the student's name, age, and contact information, and inherits from the appropriate course or degree class depending on the student's field of study and stage of progress. We could then define additional derived classes such as **ICTStudent**, **BscITStudent**, and **BscITStage1Student** as needed to capture more detailed information about the students and their progress.