|  |
| --- |
| **SOLID Principles** |
| **Question 1 –**  Which S.O.L.I.D. principle does the Employer class violate?  **Answer –**  Employer class should depend on the abstraction and not on the concretion. In other words, Employer need not explicitly know which different categories of employees it employs. |
| **Question 2 –**  Which S.O.L.I.D. principle does the following code violate?  **Answer –**  Violates the Interface Segregation principle. It is better to have Client specific interfaces rather than to have a general purpose interface. Since we have two implementations Book and DVD, GetPlayTime, GetCastList should applicable to DVD client only. So, divide the ILibraryItem interface into two interfaces IBookLibraryItem and IDVDLibraryItem. |
| **Question 3 –**  Which S.O.L.I.D. principle does the ProfitReport class violate?  **Answer –** |
| **Question 4 –**  Which S.O.L.I.D. principle does the following code violate?  **Answer –** |
| **Question 5 –**  Which S.O.L.I.D. principle does the following code violate?  **Answer –** |
| **Question 6 –**  Which S.O.L.I.D. principle does the PiggyBank class violate?  **Answer –** |
| **Question 7 –**  Which S.O.L.I.D. principle does the following code violate?  **Answer –**  Violates the Interface Segregation principle. It is better to have Client specific interfaces rather than to have a general-purpose interface. We have two Aquatic and Flying Insects, Swim() and MoveAntennae() are methods speciafically to AquaticInsects and Fly() and MoveAntennae() are methods speciafically to FlyingInsects. So, we divide the IInsectinterface into two interfaces IAquaticInsect and IFlyingInsect. |