**Question 1:**

Four principles of object oriented programing

The fundamental principles of object oriented programming are:

1. Encapsulation

2. Data abstraction

3. Polymorphism

4. Inheritance

Encapsulation: Encapsulation hides the internals of a class. It is wrapping of data and functions into a single unit called class. Encapsulation means putting data and functions that operate on data in a single unit. It is the mechanism that brings together the code and the data that it manipulates and keeps both safe from outside interference. Encapsulation hides implementation details. It ensures that the structural changes remain local. A benefit of encapsulation is that it can reduce system complexity.

Data abstraction: Abstraction hides the unnecessary details and exposing only the essential features of a particular concept or object. Through abstraction we abstract the properties of an object and keep only what we need. It highlights the properties that we need and hides others. Abstraction is used to manage complexity.

Inheritance: Inheritance is a fundamental principle of object oriented programing. Inheritance is a way to reuse code of existing objects. Classes can inherit attributes and behaviour from pre-existing class called base-class. The resulting class is called derived class. This relationship between classes lead to a hierarchy. A derived class can inherit one or more properties from base class. Properties include data variables, methods and so on.

Polymorphism: Polymorphism is a Greek word meaning the ability to take more than one form. Polymorphism refers to an operation that may exhibit different behaviour in different instances. Polymorphism allows treating objects of a base class as objects its derived class. Polymorphism exhibits strong resemblance to abstraction. Polymorphism ensures that the appropriate method of derived class is called through its base class interface.

**Question 2:**

1. Code reuse and recycling: Objects created in object oriented programing can be reused in other programing. Inheritance enhances extensibility, reusability, provides abstraction and eliminates redundant code.
2. Faster development: Reuse enables faster development since code developed for one project can be reused for future projects. This also leads to reduction in cost.
3. Maintainability: Code becomes more maintainable. Identifying objects become easier because of encapsulation.