|  |
| --- |
| Define OOPS  OOPs stands for Object Oriented Programming. |

|  |
| --- |
| What are the main features of OOPs   * Inheritance * Polymorphism * Encapsulation * Abstraction |

|  |
| --- |
| What are the major components of OOPs   * Object * Class |

|  |
| --- |
| Benefits of OOPs  Models the real world  Allows code reusability |

|  |
| --- |
| Benefits of Class  We can work with class objects by performing the following two types of operation.   * Creating attribute reference * Creating an instance of a Class |

|  |
| --- |
| Attributes  A class attribute is an element of a class  The class attributes belong to the class in which they are defined  The class attributes are following two types.   * Data attributes * Functional attributes   **Data attributes** are commonly known as **static members or class variables**  **Functional attributes or Method Class attributes** are the class methods  Methods can be invoked only by using instance of the class to which they belong |

|  |
| --- |
| \_\_init\_\_()  The \_\_init\_\_() is a constructor or a special method that can be defined in a class to create objects in the initial state  \_\_init\_\_() special methods has self as the first argument like any other function or method defined in python |

|  |
| --- |
| In how many ways class can be implemented  There are two types   1. Composition 2. Derivation   In **Composition**, classes are combined to create a code that provides better functionality  **Derivation** provides a powerful features of OOPs, which allows for the use of the features and behavior of class by another class without disturbing the rest of the program |

|  |
| --- |
| Variables  There are three different types of variables in OOPs in python.   * Instance variable ( Object level variable ) * Static variable ( Class level variable ) * Local variable ( Method level variable ) |

|  |
| --- |
| Methods  Three different types of Methods in OOPs in python.   * Instance method * Static method * Class method |

|  |
| --- |
| Reference variable:  This variable is a reference or a pointer to an object to perform operations on that object.  Reference variables always points to an object. |