Overloading

Duplicate methods/constructor names are allowed inside the same class, as long as their parameters count or declaration are different.

- **method overloading**
  - Two or more methods having the same name can be created inside a single class as long as their parameters count or declaration are different.
    - In this case, the methods are said to be overloaded and the concept is knows as Method overloading
  - Compiler error will be displayed when more than one method has the same name - Demonstrate here
  - Demonstrate how method overloading concept can avoid compiler error - Demonstrate here
- **constructor overloading**
  - The same concept of method overloading when applied to constructors is known as constructor overloading
    - In this case, the constructors are said to be overloaded and the concept is known as Constructor overloading

---

Packages

Packages are created to group related classes/interfaces/other files.

- We generally group things to organize them better for locating them easily.
- Default package - Create a new Java project say Facebook and Create a new Java Class say 'FacebookLogin' and observed that a default package will be created. - view here
- Package creation - Create a new Java project say Facebook and group the Classes under various packages - view here
- Demonstrate - Accessing instances variables and methods from other class which is under the same package
- Demonstrate - Importing the Classes in the other packages while accessing the instance variables and methods created in the Classes which are under other packages
- Demonstrate - Using * in the import statements to import all the classes in the package instead of importing a single class every time

---

Inheritance

Inheritance is a mechanism in which one class acquires the properties (i.e. variables and methods) of another class

- The purpose of this Inheritance is to use the properties (i.e. methods and variables) inside a class instead of recreating the same properties again in new class.
- Child class acquires the properties (i.e. variables and methods) of Parent Class.
- Child class uses **extends** keyword to inherit the properties from parent class
- Demonstrate a child class which inherits the properties from Parent Class - Demonstrate here
  - Child class can have specific properties (i.e. variables and methods) which are not available in the parent class
  - Object created for parent class can access the variables and methods that are created in parent class only. It cannot access the child class properties.
  - Object created for child class which is inheriting the parent class can access the variables and methods of both parent class and child class.
