Dynamic Polymorphism

- In Java, if a superclass and subclass contain methods with the same name, the version to be invoked will be decided by the JVM at runtime.
- The preceding decision to invoke the appropriate method is known as dynamic polymorphism.
- Dynamic polymorphism is implemented in Java by method overriding.
- Method overriding enables a subclass to provide its own implementation of a method that already has an implementation defined in its superclass.
- To override a method present in the superclass, the subclass method should have the same name, same parameters, and same return type as the method in the superclass.

Dynamic Polymorphism (Contd.)

- You can use the code given in the embedded document to implement dynamic polymorphism:


Microsoft Word Document

- It is important to consider the following points while implementing overriding:
  - Private methods cannot be overridden, as they are not accessible in subclasses.
  - Final methods cannot be overridden.
  - An overridden method cannot be granted more restrictive access rights in a subclass than it is assigned in case of a superclass.

Just a minute

- What will be the output of the following code?

```
public class MethodDemo
{
void print()
{
   System.out.println("Print1");
}
void print(String a)
{
   System.out.println("Print2");
}
String print()
{
   System.out.println("Print3");
   return "Print3";
```

Just a minute (Contd.)

```
}
public static void main(String args[])
  {
      MethodDemo obj=new MethodDemo();
      obj.print();
  }
}
```

- Print1
- Compile-time error
- Print2
- Print3

Just a minute (Contd.)

- Solution:
  - Compile-time error

Activity 5.1: Implementing Inheritance and Polymorphism

- Problem Statement:
  - The management of LearnMore University is planning to automate the University management system. Therefore, Steve Wilkinson, the programmer, has decided to create a Java program that accepts the student details, such as the first name, last name, age, course enrolled, and student ID. In addition, he also needs to accept the employee details, such as first name, last name, age, salary, department name, designation, and employee ID. Steve must ensure the reusability of code. The program must offer a choice to accept either the student's or employee's details. Help Steve to develop the program.