KEMENTERIAN PENDIDIKAN TINGGI

POLITEKNIK MALAYSIA
METrO Tasek Gelugor

Politeknik METrO Tasek Gelugor

NO 25, LORONG KOMERSIAL 2,

PUSAT KOMERSIAL TASEK GELUGOR

13300 TASEK GELUGOR

PULAU PINANG

DFP30243
OBJECT ORIENTED PROGRAMMING

NAME:	MUHAMMAD AFIQ MUHAIMIN BIN MOHD ZAINI
CLASS:	DDT3A
REGISTRATION NO:	32DDT20F2029
LECTURER:	**PUAN HAZLEENA BINTI OSMAN** **PUAN RODZIAH BINTI IBRAHIM**
NAME OF TASK:	LAB ACTIVITY 5

LAB ACTIVITY 5:

Method and Constructor Overloading

Learning Outcomes

This Lab sheet encompasses 6 activities (Activity 5A until 5C).

By the end of this lab, students should be able to:

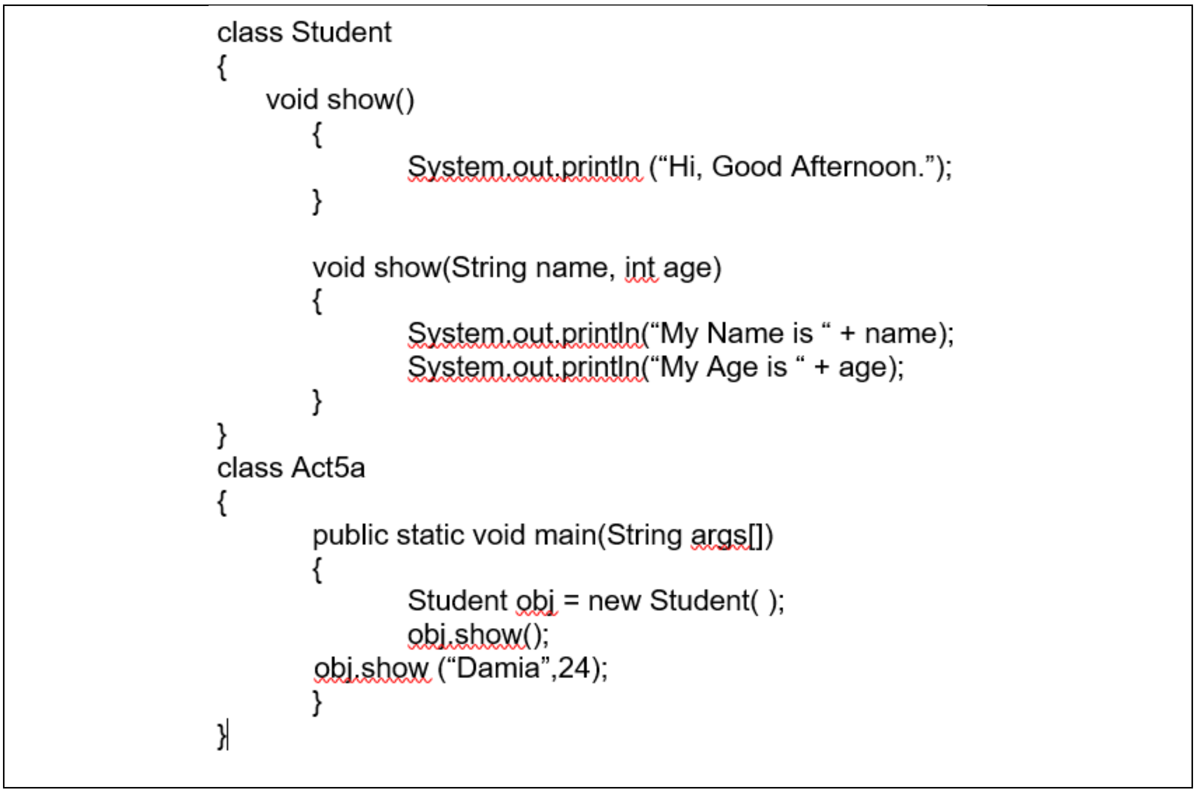
- **Perform constructor and constructor overloading in Java programs.**

Activity 5A

Activity Outcome: Understand and implement Method Overloading

Procedure:

Step 1: Type the programs given below

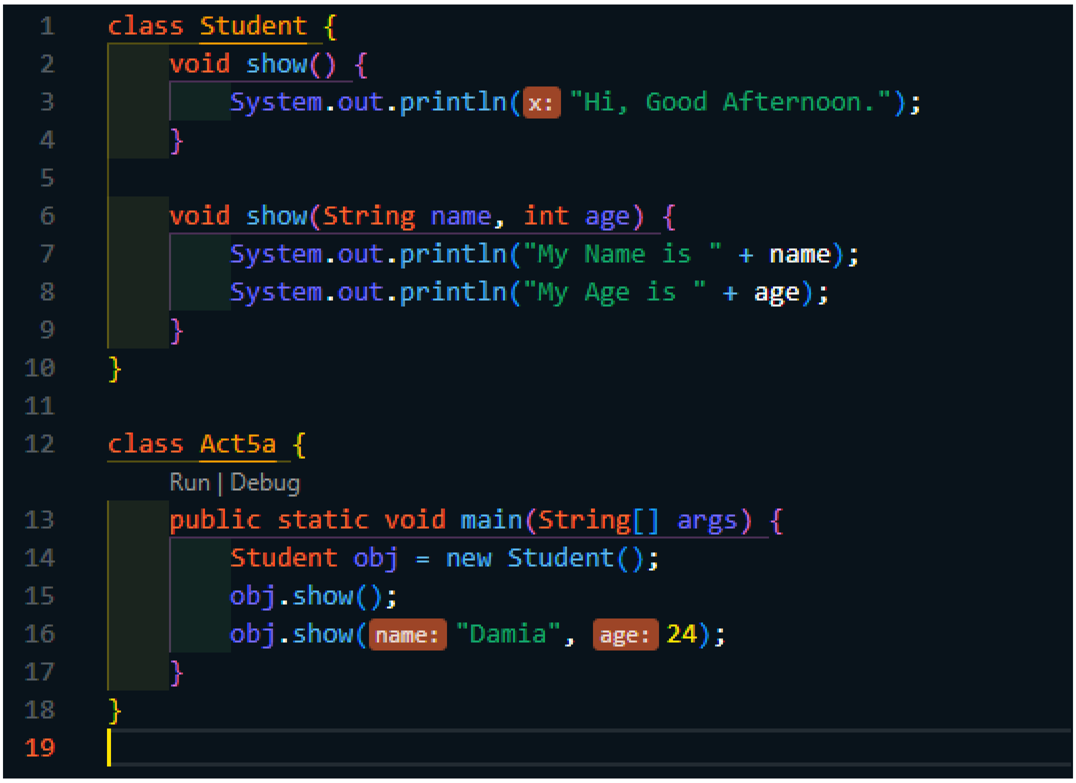


```
class Student
{
    void show()
        {
                System.out.println (“Hi, Good Afternoon.”);
        }

        void show(String name, int age)
        {
                System.out.println(“My Name is “ + name);
                System.out.println(“My Age is “ + age);
        }
}
class Act5a
{
        public static void main(String args[])
        {
                Student obj = new Student( );
                obj.show();
        obj.show (“Damia”,24);
        }
}
```

Step 2: Save the program as `Act5a.java`

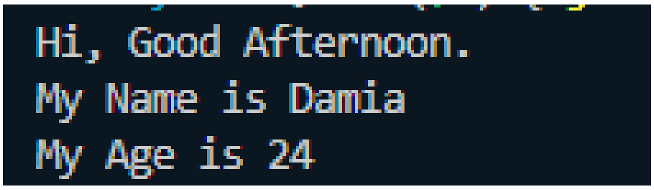
Step 3: Compile and run the program. Observe the output.

Code



```java
class Student {
    void show() {
        System.out.println(x: "Hi, Good Afternoon.");
    }

    void show(String name, int age) {
        System.out.println("My Name is " + name);
        System.out.println("My Age is " + age);
    }
}

class Act5a {
    Run | Debug
    public static void main(String[] args) {
        Student obj = new Student();
        obj.show();
        obj.show(name: "Damia", age: 24);
    }
}

```

Output



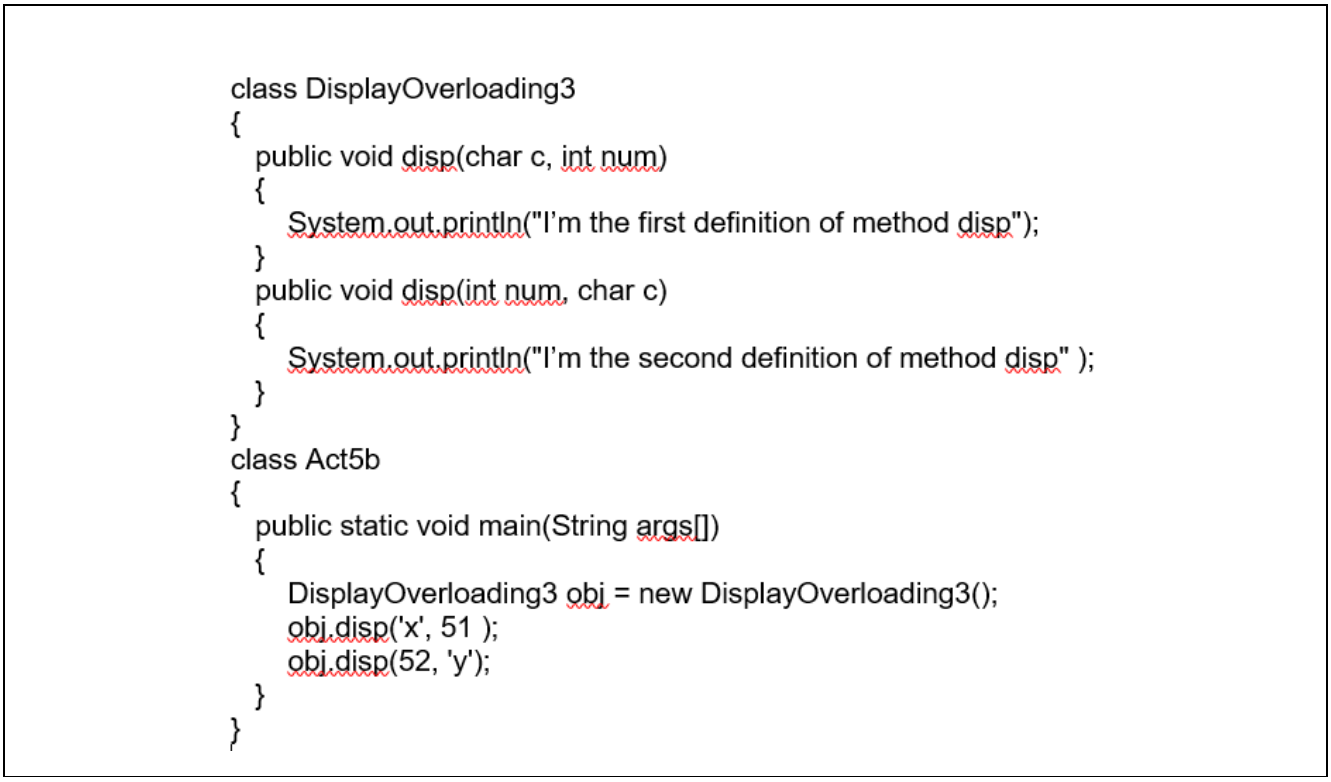
```
Hi, Good Afternoon.
My Name is Damia
My Age is 24
```

Activity 5B

Activity Outcome: Understand and create Method Overloading.

Procedure:

Step 1: Type the programs given below

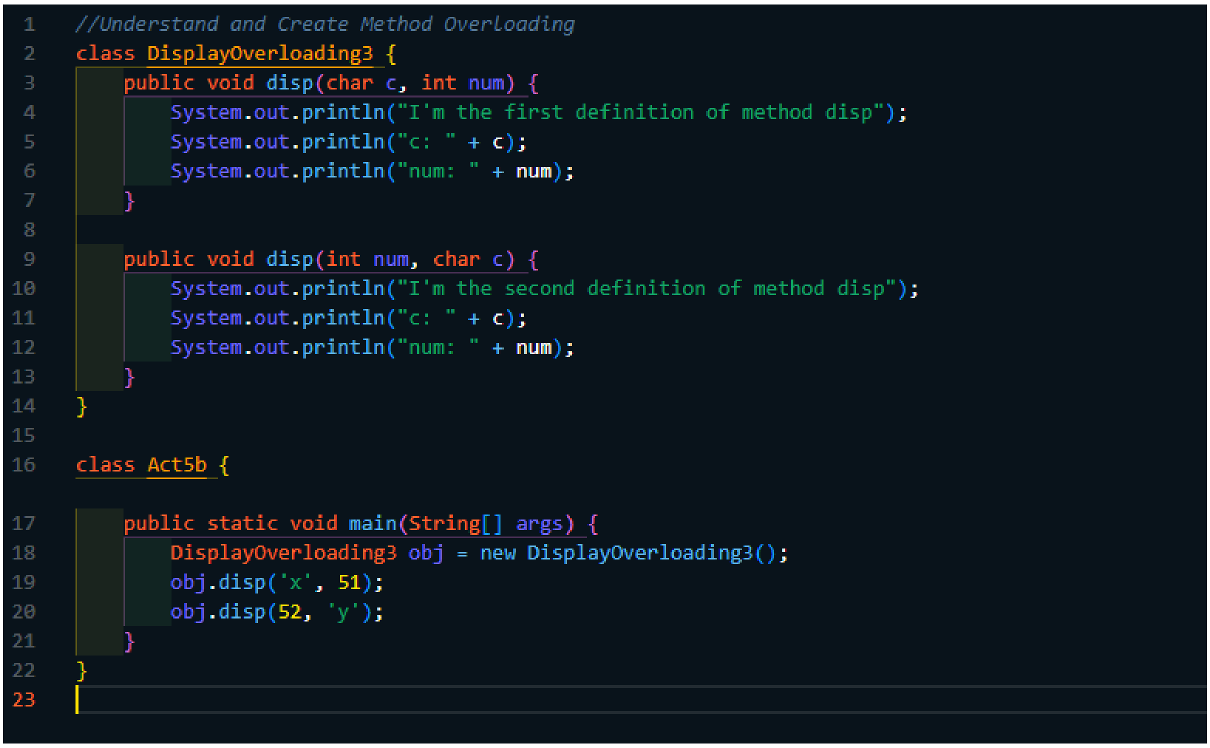


```
class DisplayOverloading3
{
  public void disp(char c, int num)
  {
    System.out.println("I'm the first definition of method disp");
  }
  public void disp(int num, char c)
  {
    System.out.println("I'm the second definition of method disp" );
  }
}
class Act5b
{
  public static void main(String args[])
  {
    DisplayOverloading3 obj = new DisplayOverloading3();
    obj.disp('x', 51 );
    obj.disp(52, 'y');
  }
}
```

Step 2: Save the program as `Act5b.java`

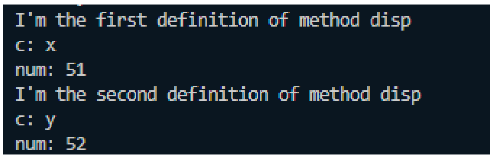
Step 3: Compile and run the program. Observe the output.

Code



```java
//Understand and Create Method Overloading
class DisplayOverloading3 {
    public void disp(char c, int num) {
        System.out.println("I'm the first definition of method disp");
        System.out.println("c:  " + c);
        System.out.println("num:  " + num);
    }

    public void disp(int num, char c) {
        System.out.println("I'm the second definition of method disp");
        System.out.println("c:  " + c);
        System.out.println("num:  " + num);
    }
}

class Act5b {
    public static void main(String[] args) {
        DisplayOverloading3 obj = new DisplayOverloading3();
        obj.disp('x', 51);
        obj.disp(52, 'y');
    }
}

```

Output



```
I'm the first definition of method disp
c: x
num: 51
I'm the second definition of method disp
c: y
num: 52
```

Activity 5C

Activity Outcome: Understand and create constructor.

Procedure:

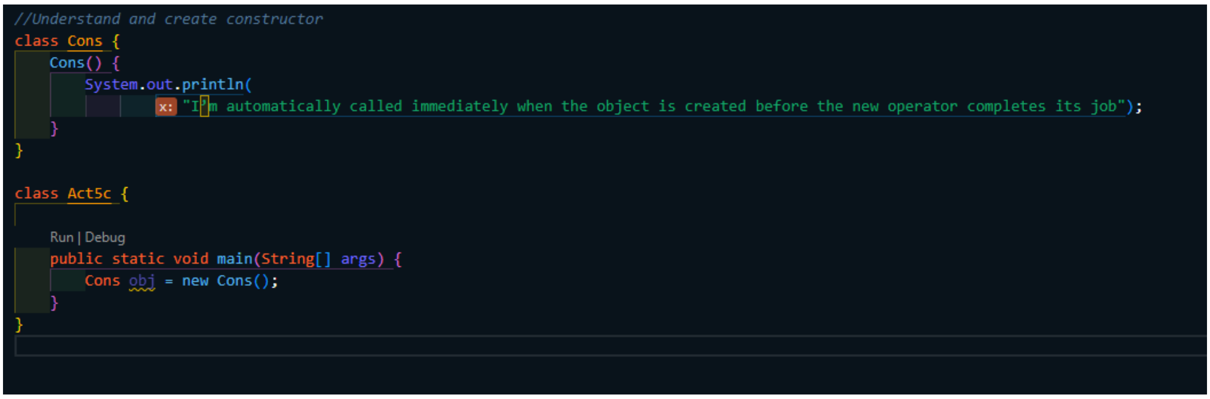
Step 1: Type the programs given below

```
class Cons

{

      Cons()
      {

            System.out.println ("I'm automatically called immediately when the

            object is created before the new operator completes its job");

      }

}


class Act5c

{

      public static void main(String args[])

      {

            Cons obj = new Cons();

      }

}
```

Step 2: Save the program as `Act5c.java`

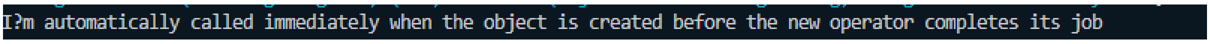
Step 3: Compile and run the program. Observe the output.

Code



```
//Understand and create constructor
class Cons {
    Cons() {
        System.out.println(
                x: "I’m automatically called immediately when the object is created before the new operator completes its job");
    }
}

class Act5c {

    Run | Debug
    public static void main(String[] args) {
        Cons obj = new Cons();
    }
}
```

Output



```
I?m automatically called immediately when the object is created before the new operator completes its job
```

Activity 5D

Activity Outcome: Understand and create constructor overloading.

Procedure:

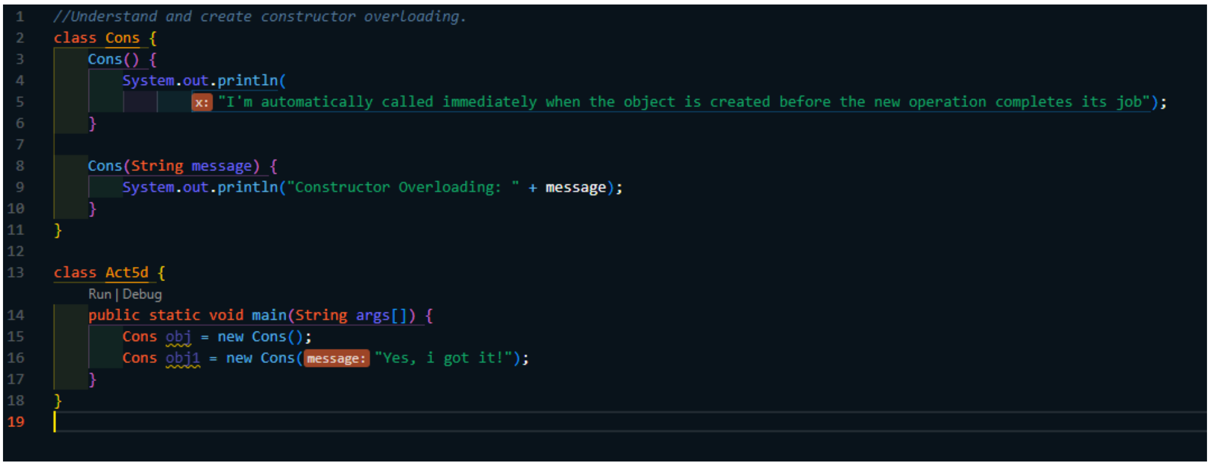
Step 1: Type the programs given below

```
class Cons

{

       Cons()
       {

              System.out.println (“I’m automatically called immediately when

              the object is created before the new operator completes its

              job”);

       }


       Cons(String message)

       {

              System.out.println(“Constructor Overloading” + message);

       }

}


class Act5d

{

       public static void main(String args[])

       {

              Cons obj = new Cons();

              Cons obj = new Cons(“Yes, I got it!”);

       }

}
```

Step 2: Save the program as `Act5d.java`

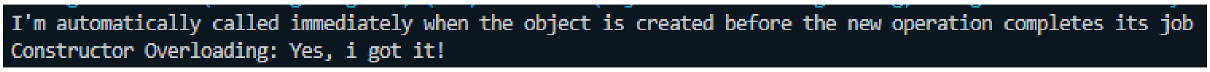
Step 3: Compile and run the program. Observe the output.

Code



```java
//Understand and create constructor overloading.
class Cons {
    Cons() {
        System.out.println(
                "I'm automatically called immediately when the object is created before the new operation completes its job");
    }

    Cons(String message) {
        System.out.println("Constructor Overloading: " + message);
    }
}

class Act5d {
    public static void main(String args[]) {
        Cons obj = new Cons();
        Cons obj1 = new Cons("Yes, i got it!");
    }
}
```

Output



```
I'm automatically called immediately when the object is created before the new operation completes its job
Constructor Overloading: Yes, i got it!
```