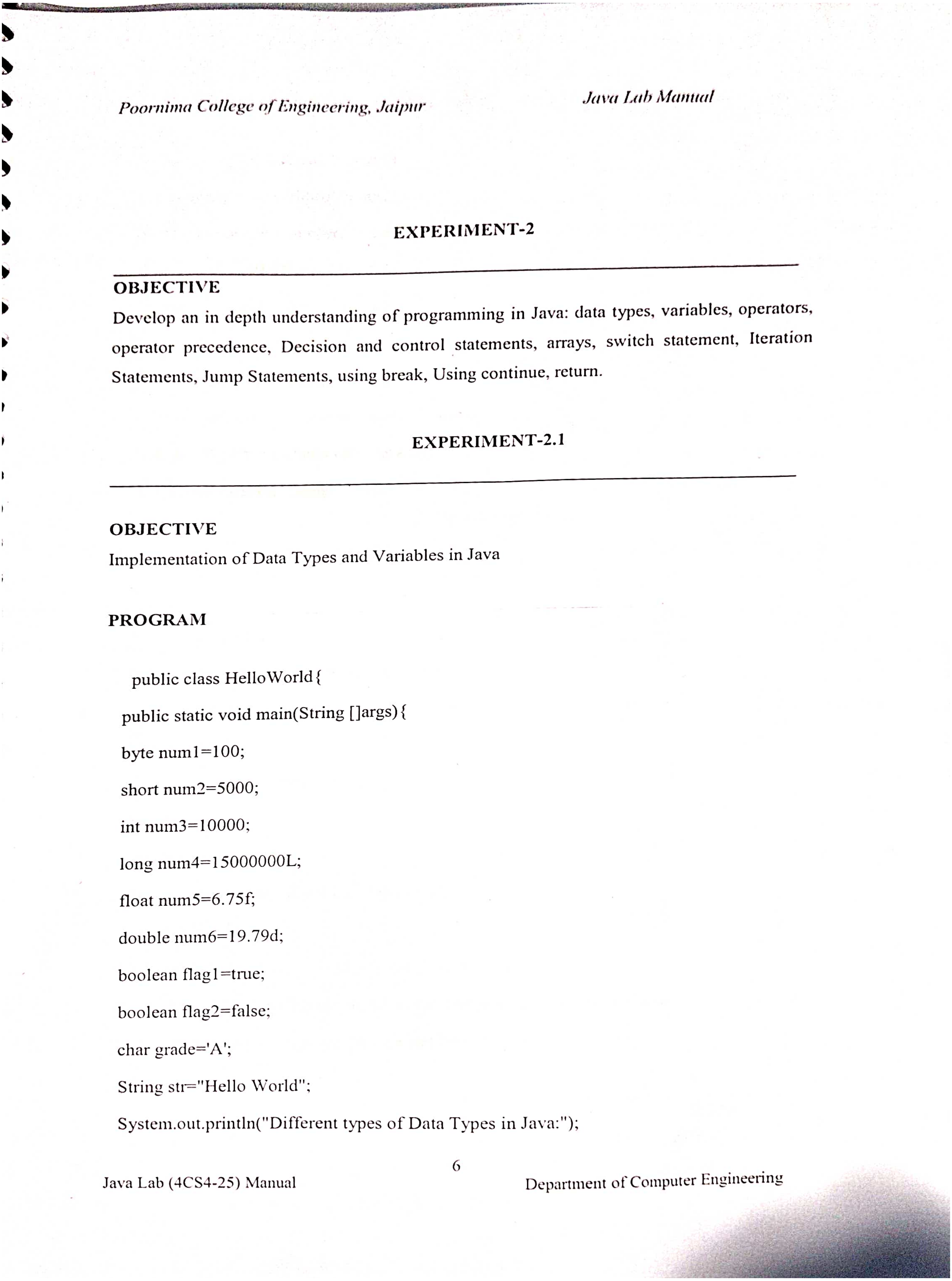


EXPERIMENT-2

OBJECTIVE

Develop an in depth understanding of programming in Java: data types, variables, operators, operator precedence, Decision and control statements, arrays, switch statement, Iteration Statements, Jump Statements, using break, Using continue, return.

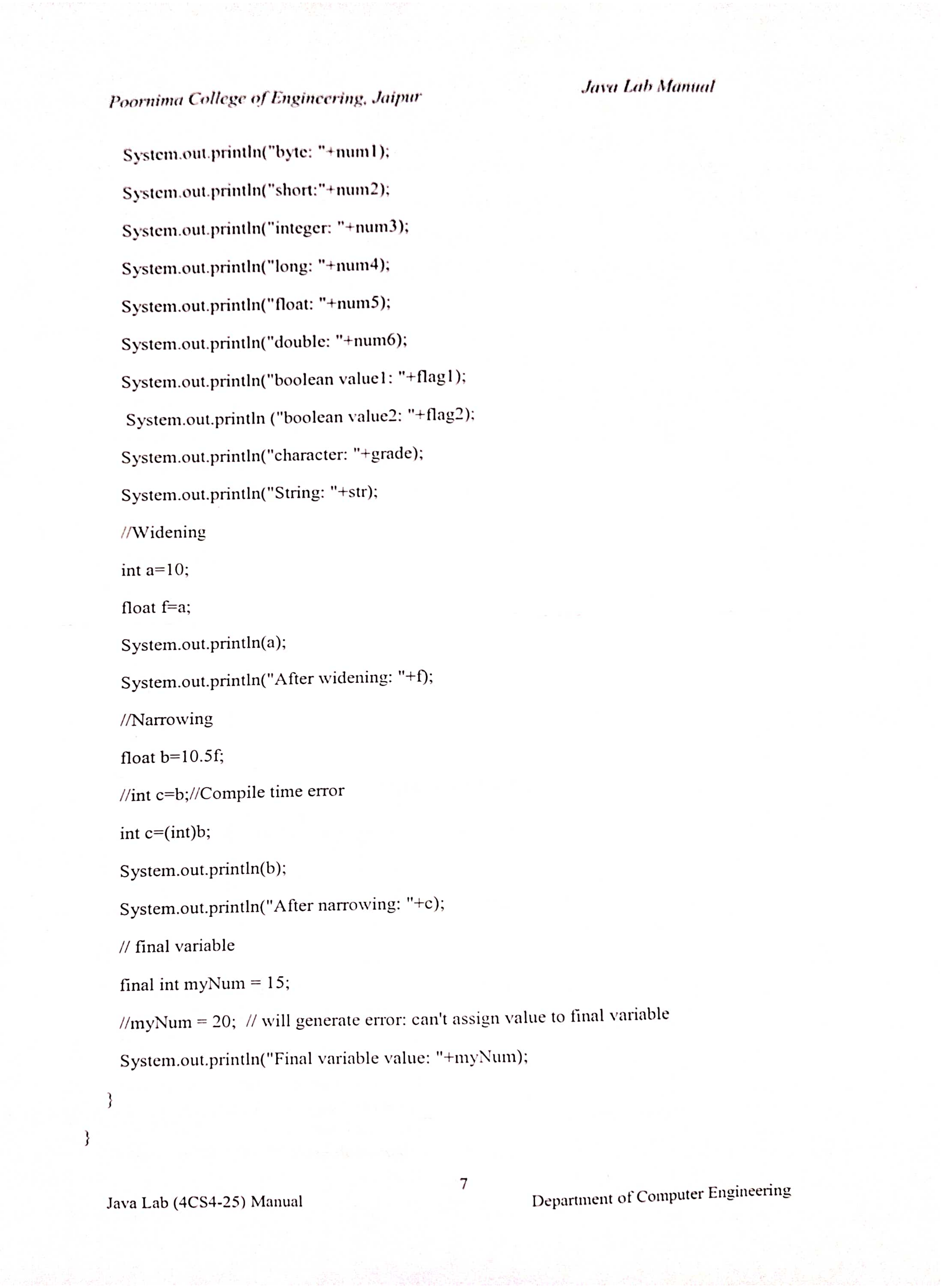
EXPERIMENT-2.1

OBJECTIVE

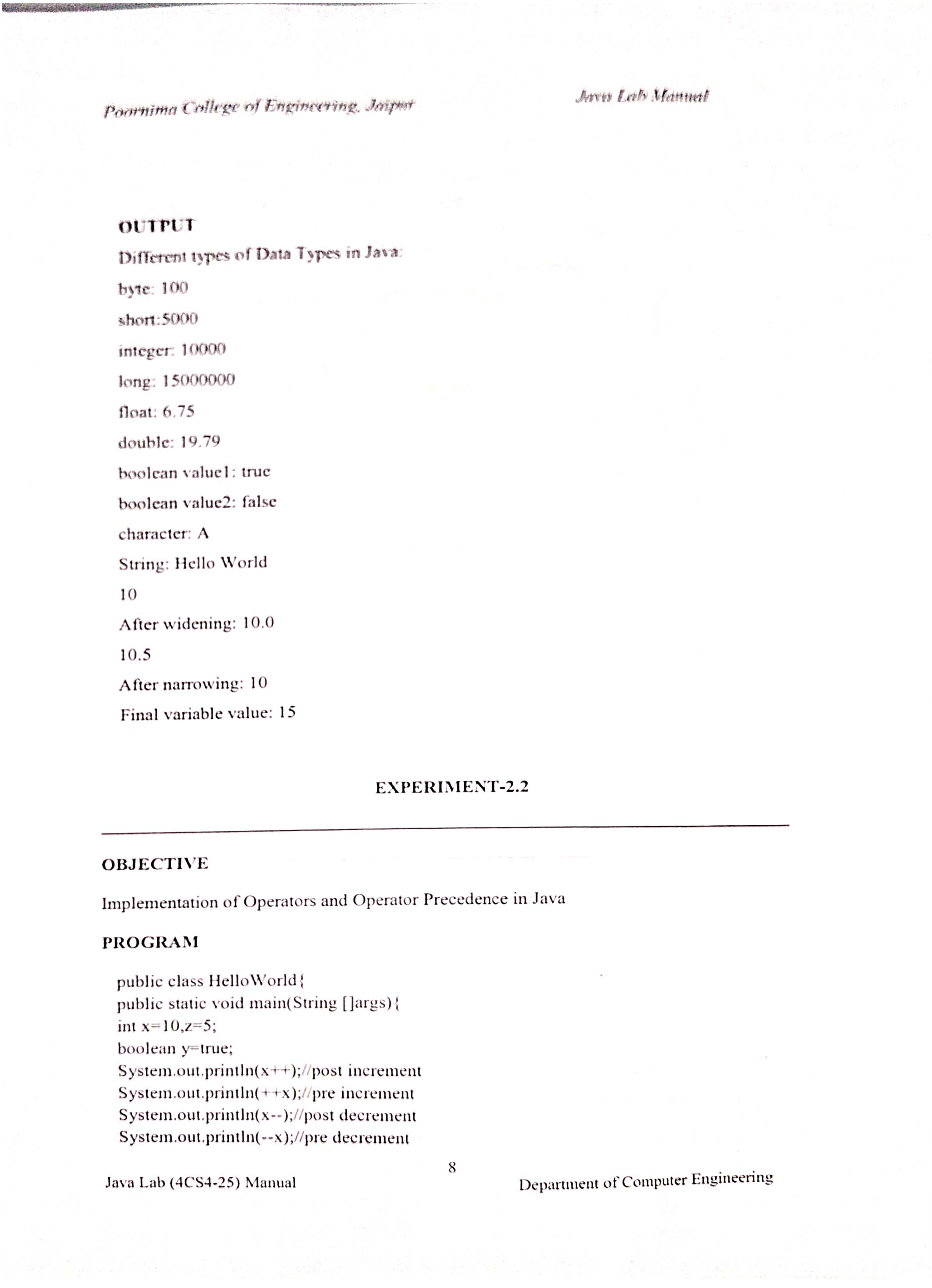
Implementation of Data Types and Variables in Java

PROGRAM

```
  public class HelloWorld{
public static void main(String []args){
byte num1=100;
short num2=5000;
int num3=10000;
long num4=15000000L;
float num5=6.75f;
double num6=19.79d;
boolean flag1=true;
boolean flag2=false;
char grade='A';
String str="Hello World";
System.out.println("Different types of Data Types in Java:");
```



```
    System.out.println("byte: "+num1);
    System.out.println("short:"+num2);
    System.out.println("integer: "+num3);
    System.out.println("long: "+num4);
    System.out.println("float: "+num5);
    System.out.println("double: "+num6);
    System.out.println("boolean value1: "+flag1);
     System.out.println ("boolean value2: "+flag2);
    System.out.println("character: "+grade);
    System.out.println("String: "+str);
    //Widening
    int a=10;
    float f=a;
    System.out.println(a);
    System.out.println("After widening: "+f);
    //Narrowing
    float b=10.5f;
    //int c=b;//Compile time error
    int c=(int)b;
    System.out.println(b);
    System.out.println("After narrowing: "+c);
    // final variable
    final int myNum = 15;
    //myNum = 20;  // will generate error: can't assign value to final variable
    System.out.println("Final variable value: "+myNum);
  }
}
```



OUTPUT

```
Different types of Data Types in Java:
byte: 100
short:5000
integer: 10000
long: 15000000
float: 6.75
double: 19.79
boolean value1: true
boolean value2: false
character: A
String: Hello World
10
After widening: 10.0
10.5
After narrowing: 10
Final variable value: 15
```

EXPERIMENT-2.2

OBJECTIVE

Implementation of Operators and Operator Precedence in Java

PROGRAM

```
public class HelloWorld{
public static void main(String []args){
int x=10,z=5;
boolean y=true;
System.out.println(x++);//post increment
System.out.println(++x);//pre increment
System.out.println(x--);//post decrement
System.out.println(--x);//pre decrement
```

```
        System.out.println(~x);//negation
        System.out.println(!y);//inverting boolean value
        System.out.println("Addition:"+(x+z));//addition
        System.out.println("Subtraction:"+(x-z));//subtraction
        System.out.println("Division:"+(x/z));//division
        System.out.println("Multiplication:"+(x*z));// multiplication
        System.out.println("Modulus:"+(x%z));// modulus
        System.out.println(x<<1);//left shift
        //For positive number, >> and >>> works same
        System.out.println(x>>1);//signed right shift
        System.out.println(x>>>1);// unsigned right shift
        System.out.println(-x>>1);
        System.out.println(-x>>>1);
        System.out.println(x<=z);//relational operator
        System.out.println(x!=z);//relational operator
        System.out.println(x<15 && z>20);//logical AND
        System.out.println(x<15 || z>20);//logical OR
        System.out.println(x&z);//bitwise AND
        System.out.println(x|z);//bitwise OR
        System.out.println(x);//signed right shift
        System.out.println(x>>1);// unsigned right shift
        System.out.println((x<z)?x:z); // ternary operator for minimum no
        x+=5; // assignment operator
        z*=2; // assignment operator
        System.out.println(x);
        System.out.println(z);
        // Operator Precedence
        int result= x - ++z -z++;
        System.out.println("Result is:"+result);
        double fahrenheit = 98.4;
        double celsius  = ( 5.0 * (fahrenheit - 32.0) ) / 9.0;// operator precedence
        System.out.println("Celsius: "+celsius);
    }
}
```

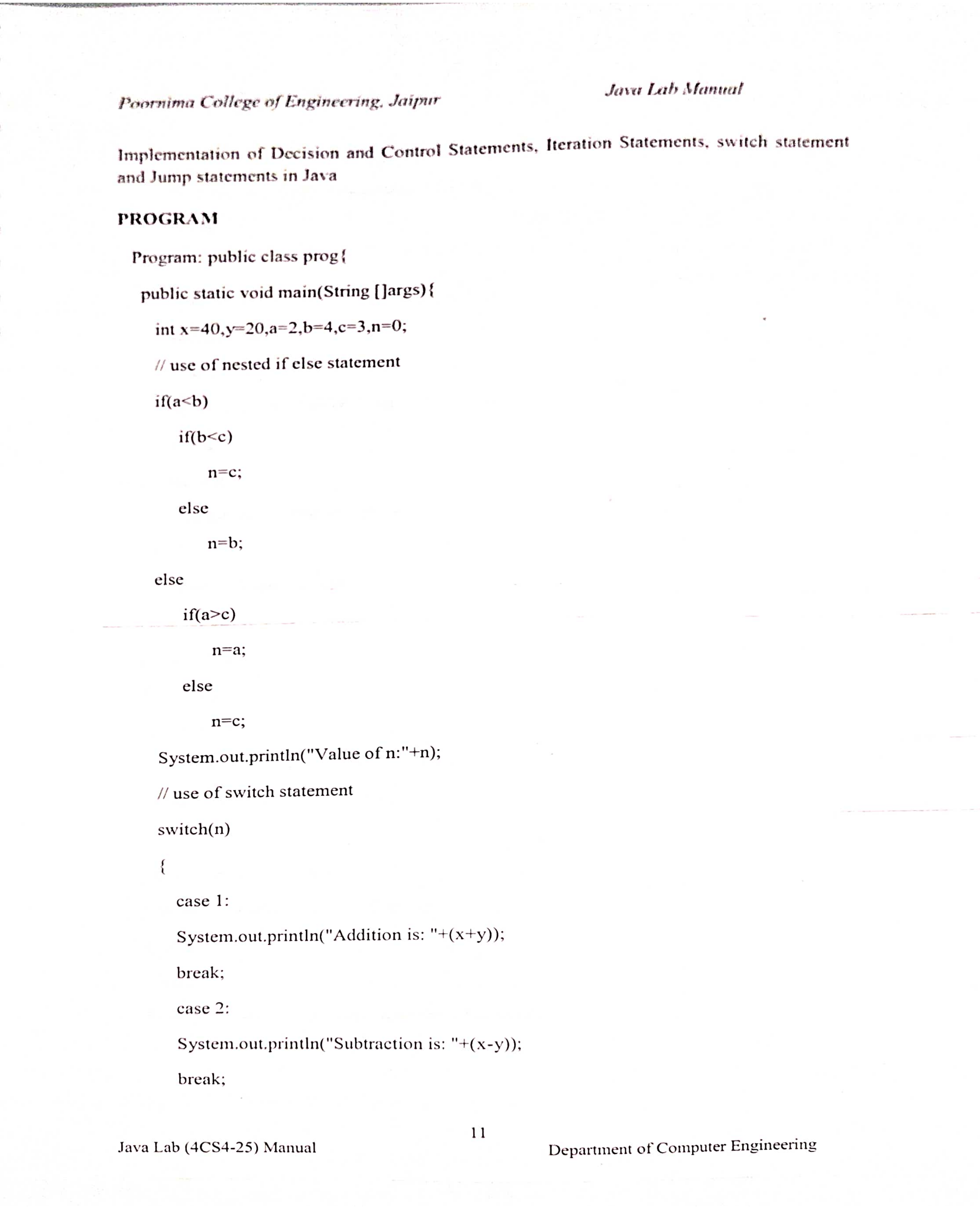
OUTPUT

```
10
12
12
10
-11
false
Addition:15
```

```
Subtraction:5
Division:2
Multiplication:50
Modulus:0
20
5
5
-5
2147483643
false
true
false
true
0
15
10
5
5
15
10
Result is:-7
Celsius: 36.888888888888886
```

EXPERIMENT-2.3

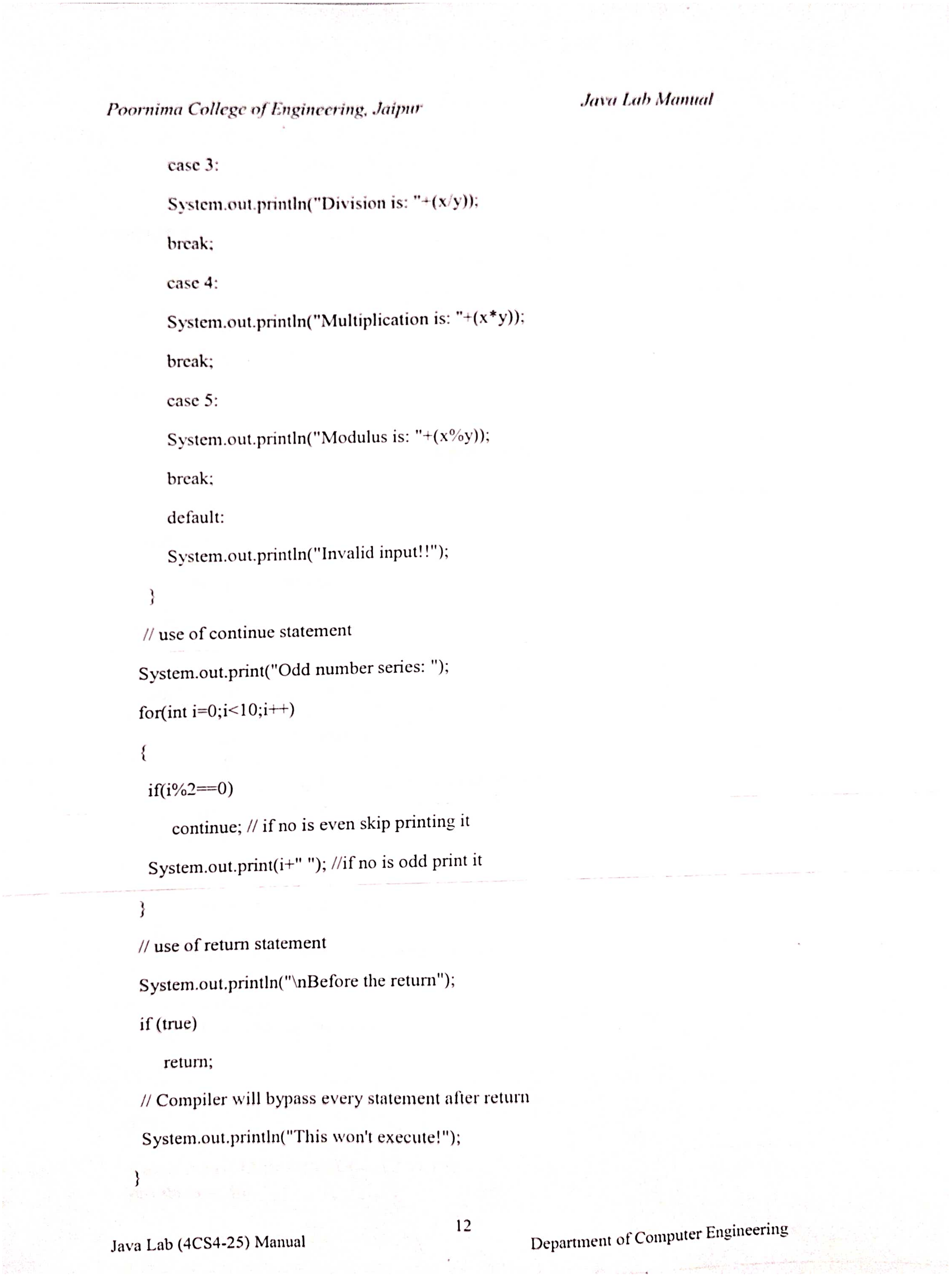
OBJECTIVE



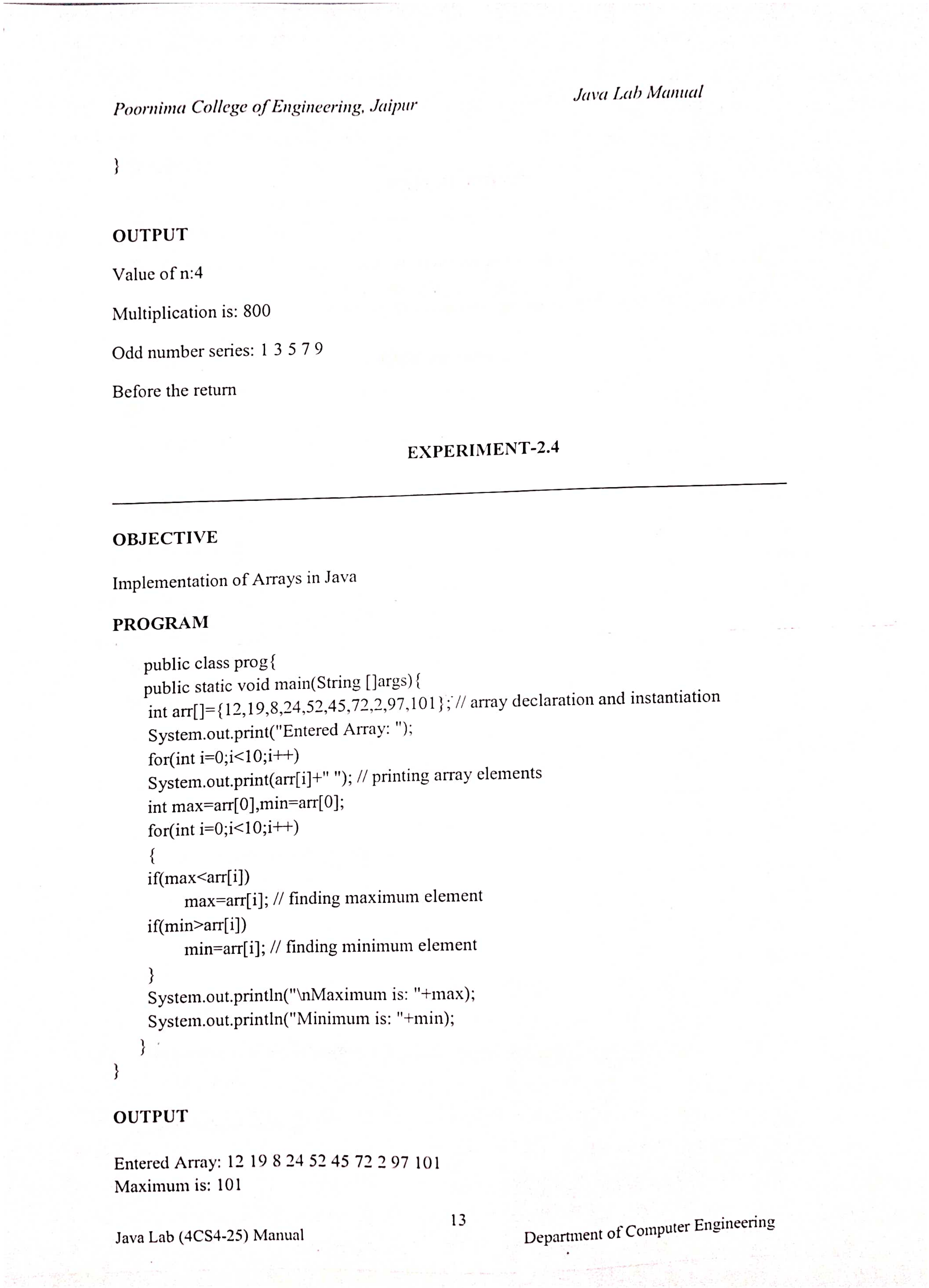
Implementation of Decision and Control Statements, Iteration Statements, switch statement and Jump statements in Java

PROGRAM

```
Program: public class prog{
  public static void main(String []args){
    int x=40,y=20,a=2,b=4,c=3,n=0;
    // use of nested if else statement
    if(a<b)
        if(b<c)
             n=c;
        else
             n=b;
    else
        if(a>c)
             n=a;
        else
             n=c;
    System.out.println("Value of n:"+n);
    // use of switch statement
    switch(n)
    {
       case 1:
       System.out.println("Addition is: "+(x+y));
       break;
       case 2:
       System.out.println("Subtraction is: "+(x-y));
       break;
```



```
        case 3:
        System.out.println("Division is: "+(x/y));
        break;
        case 4:
        System.out.println("Multiplication is: "+(x*y));
        break;
        case 5:
        System.out.println("Modulus is: "+(x%y));
        break;
        default:
        System.out.println("Invalid input!!");
     }
    // use of continue statement
    System.out.print("Odd number series: ");
    for(int i=0;i<10;i++)
    {
     if(i%2==0)
        continue; // if no is even skip printing it
     System.out.print(i+" "); //if no is odd print it
    }
    // use of return statement
    System.out.println("\nBefore the return");
    if (true)
       return;
    // Compiler will bypass every statement after return
    System.out.println("This won't execute!");
   }
```



```
}
```

OUTPUT

```
Value of n:4

Multiplication is: 800

Odd number series: 1 3 5 7 9

Before the return
```

EXPERIMENT-2.4

OBJECTIVE

Implementation of Arrays in Java

PROGRAM

```
    public class prog{
    public static void main(String []args){
     int arr[]={12,19,8,24,52,45,72,2,97,101}; // array declaration and instantiation
     System.out.print("Entered Array: ");
     for(int i=0;i<10;i++)
     System.out.print(arr[i]+" "); // printing array elements
     int max=arr[0],min=arr[0];
     for(int i=0;i<10;i++)
     {
     if(max<arr[i])
            max=arr[i]; // finding maximum element
     if(min>arr[i])
            min=arr[i]; // finding minimum element
     }
     System.out.println("\nMaximum is: "+max);
     System.out.println("Minimum is: "+min);
   }
}
```

OUTPUT

```
Entered Array: 12 19 8 24 52 45 72 2 97 101
Maximum is: 101
```

Minimum is: 2

EXPERIMENT-3

OBJECTIVE

Write Object Oriented programs in Java: Objects, Classes constructors, returning and passing objects as parameter, Inheritance, Access Control, using super, final with inheritance Overloading and overriding methods, Abstract classes, Extended classes

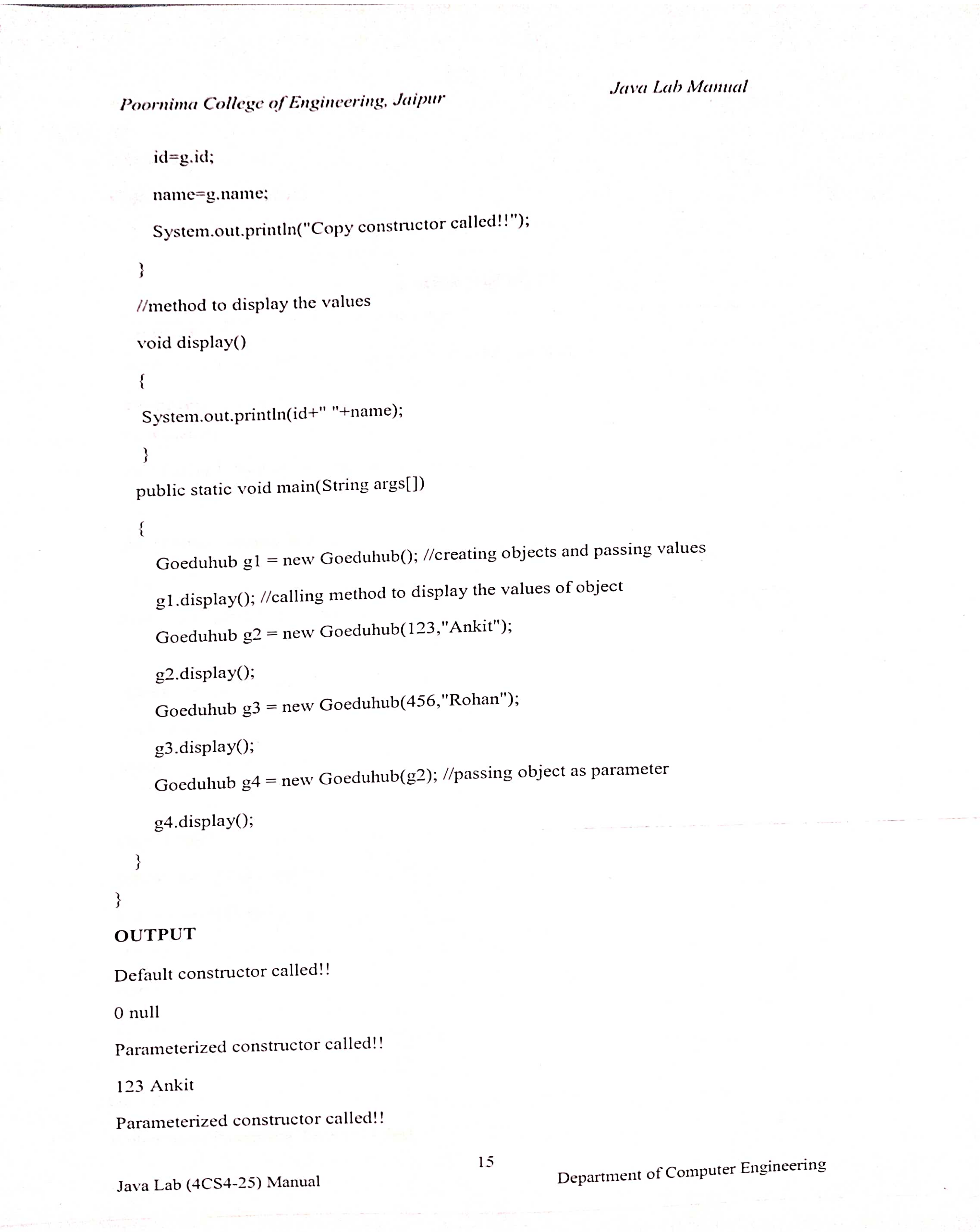
EXPERIMENT-3.1

OBJECTIVE

Implementation of Objects, Classes and Constructors in Java.

PROGRAM

```
class Goeduhub // creating a class
{
int id;
String name;
Goeduhub()   // creating default constructor
{
  System.out.println("Default constructor called!!");
}
//creating a parameterized constructor
Goeduhub(int i,String n)
{
   id = i;
   name = n;
   System.out.println("Parameterized constructor called!!");
}
Goeduhub(Goeduhub g)
{
```



```
        id=g.id;

        name=g.name;

        System.out.println("Copy constructor called!!");

    }

    //method to display the values

    void display()

    {

     System.out.println(id+" "+name);

     }

    public static void main(String args[])

    {

        Goeduhub g1 = new Goeduhub(); //creating objects and passing values

        g1.display(); //calling method to display the values of object

        Goeduhub g2 = new Goeduhub(123,"Ankit");

        g2.display();

        Goeduhub g3 = new Goeduhub(456,"Rohan");

        g3.display();

        Goeduhub g4 = new Goeduhub(g2); //passing object as parameter

        g4.display();

    }

}
```

OUTPUT

```
Default constructor called!!

0 null

Parameterized constructor called!!

123 Ankit

Parameterized constructor called!!
```

```
456 Rohan

Copy constructor called!!

123 Ankit
```

EXPERIMENT-3.2

OBJECTIVE

Implementation of Inheritance and Access control in Java

Program:

```
class Iname{

void fun1() {  System.out.println("Technologies !");

} }

class fname extends Iname{

void fun2() {

System.out.print("Poornima ");}

}

class Greet extends fname{

void fun3(){

System.out.print("Welcome to ");}

}

class Test1{

public static void main(String args[]){

Greet d=new Greet();

d.fun3();

d.fun2();

d.fun1();

}}
```

OUTPUT

```
Welcome to Poornima Technologies!
```

EXPERIMENT-3.3

OBJECTIVE

Implementation of super and final keywords in Java

Program: // superclass Person

```
class Person
{
   int id=111;
   void message()
   {
    System.out.println("Welcome to Goeduhub!");
   }
  Person()
   {
     System.out.println("Person class Constructor");
   }
}
// subclass Student extending the Person class
class Student extends Person // Inheritance
{
   Student()
   {
   super(); // invoke or call parent class constructor
  System.out.println("Student class Constructor");
  }
  void message()
```

```
 {
  System.out.println("Technologies");
  }
  void display()
  {
 super.message(); // calling super class method
 message();
System.out.println("Student Id: "+super.id); //accessing super class variable
 }
}
class Test
{
 public static void main(String[] args)
 {
   Student s = new Student();
   s.display();
 }
}
```

OUTPUT

```
Person class Constructor
Student class Constructor
Welcome to Poornima!
Technologies
Student Id: 111
```