CLASS AND OBJECT

```
class Account {
  private double balance = 500.00;  // member data
  public double  getBalance(int x) {     // member method
    // logic here
    return balance;
  }

  public static void main(String[] args) {
    Account accnt = new Account();           // object creation
    double value = accnt.getBalance(123456);
    System.out.println("The balance is: " + value);
  }
}
```

OUTPUT:

```
Output:
The balance is: 500.0
```

KEYWORD AND IDENTFIER

```
class ScopeOfVariables {
  int i = 34;   // instance variable
  static int z; // class variable
  static {
    z = 10;
    System.out.println("inside static block: " + z);
  }
  void test() {
```

```
    int k = 200;  // local variable
    System.out.println("local variable: " + (k + i));
  }
  public static void main(String[] args) {
    ScopeOfVariables obj = new ScopeOfVariables();
    obj.test();
    System.out.println(obj.i);
  }
}
```

OUTPUT

```
java -cp /tmp/lmEDAeYsKp ScopeOfVariables
inside static block: 10
local variable: 234
34
```

SHORT CIRCUIT OPERATION

```
class Account {
  public static void main(String[] args) {
    double balance = 600;
    System.out.println("Amount to withdraw");
    double amount = 1500;
    if(amount < 0 || amount > balance) {
      System.out.println("Withdrawal has failed");
    }
    else {
      balance -= amount;
      System.out.println("Withdrawal has succeeded");
    }
  }
}
```

OUTPUT

```
java -cp /tmp/lmEDAeYsKp Account
Amount to withdraw
Withdrawal has failed
```

SHORT CIRCUIT AND LOGIC

```
class Account {
  public static void main(String[] args) {
    double balance = 2000;
    System.out.println("Amount to withdraw");
    double amount = 500, limit = 10000, minbal = 500;
    if(amount <= limit && (balance - amount) > minbal) {
      balance -= amount;
      System.out.println("Withdrawal has succeeded");
    }
  }
}
```

```
java -cp /tmp/lmEDAeYsKp Account
Amount to withdraw
Withdrawal has succeeded
```

EXPLICIT TYPE CASTING

```
class Calculation{
    public static void main(String[] args){
        double d = 234.04;
        long l = (long)d;   //explicit type casting
```

```
        int i = (int)l;     //explicit type casting
        System.out.println("double value " + d);
        System.out.println("long value " + l);
        System.out.println("int value " + i);
    }
}
```

OUTPUT

```
java -cp /tmp/lmEDAeYsKp Calculation
double value 234.04
long value 234int value 234
```

IMPLICIT TYPE CASTING

```
class Calculation{
    public static void main (String[] args) {
        int i = 300;
        long l = i;     //no explicit type casting
        float f = l;    //no explicit type casting
        System.out.println("int value " + i);
        System.out.println("long value " + l);
        System.out.println("float value " + f);
    }
}
```

OUTPUT

```
int value 300
long value 300
float value 300.0
```

EXERCISE

```
public class Student {
    private String name;
    private int studentID;
    private char residentialStatus;

    // Getter and Setter for name
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    // Getter and Setter for studentID
    public int getStudentID() {
        return studentID;
    }

    public void setStudentID(int studentID) {
        this.studentID = studentID;
    }

    // Getter and Setter for residentialStatus
    public char getResidentialStatus() {
        return residentialStatus;
    }

    public void setResidentialStatus(char residentialStatus) {
        // Validate the input for residential status
```

```
        if (residentialStatus == 'H' || residentialStatus == 'D') {
            this.residentialStatus = residentialStatus;
        } else {
            System.out.println("Invalid residential status. Please enter 'H' for hosteller or 'D' for day scholar.");
        }
    }

    public static void main(String[] args) {
        // Creating an instance of the Student class
        Student student = new Student();

        // Setting student details using setter methods
        student.setName("John Doe");
        student.setStudentID(123456);
        student.setResidentialStatus('H'); // 'H' for hosteller

        // Getting student details using getter methods and displaying them
        System.out.println("Student Name: " + student.getName());
        System.out.println("Student ID: " + student.getStudentID());
        System.out.println("Residential Status: " + student.getResidentialStatus());
    }
}
```

OUTPUT:

Student Name: John Doe

Student ID: 123456

Residential Status: H

EXERCISE-2

```
public class Rectangle {
```

```
    private int length;
    private int breadth;

    // Setter method for length
    public void setLength(int length) {
        this.length = length;
    }

    // Setter method for breadth
    public void setBreadth(int breadth) {
        this.breadth = breadth;
    }

    public static void main(String[] args) {
        Rectangle myRectangle = new Rectangle();

        // Initializing the instance variables using setter methods
        myRectangle.setLength(5);
        myRectangle.setBreadth(10);

        // Displaying the values of length and breadth
        System.out.println("Length: " + myRectangle.length);
        System.out.println("Breadth: " + myRectangle.breadth);
    }
}
```

OUTPUT:

Length: 5

Breadth: 10

```
public class Rectangle {
```

```
    private int length;
    private int breadth;

    // Setter methods for length and breadth
    public void setLength(int length) {
        this.length = length;
    }

    public void setBreadth(int breadth) {
        this.breadth = breadth;
    }

    // Method to calculate perimeter
    public void calculatePerimeter() {
        int perimeter = 2 * (length + breadth);
        System.out.println("Perimeter of the rectangle: " + perimeter);
    }
}

public class PerimeterCalculator {
    public static void main(String[] args) {
        Rectangle myRectangle = new Rectangle();

        // Initializing the instance variables using setter methods
        myRectangle.setLength(5);
        myRectangle.setBreadth(10);

        // Calculating and displaying the perimeter
        myRectangle.calculatePerimeter();
    }
}
```

OUTPUT:

Perimeter of the rectangle: 30