Labsheet 4 – JAVA

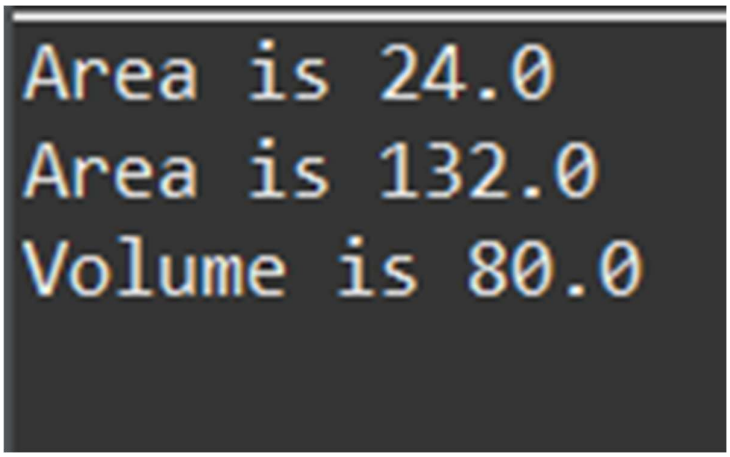
1. Create a class 'Rectangle' that contains 'length' and 'width' as data members.
   - The members of the class 'Rectangle' should consist of constructors and a method area(), which returns area of rectangle.

   From Rectangle class derive a class named 'Box' that has additional data member 'depth'.
   - The 'Box' should have constructors to initialise data members, area() that returns surface area(2*l*w+2*l*h+2*w*h) and volume() that returns volume of box.

   Write a Main class to instantiate both classes and call its methods. (Create objects of both rectangle and box classes and call all its methods.)

```java
package Questions;
class Rectangle{
    double length, width;
    Rectangle(){

    }
    Rectangle(double a, double b){
        length=a;
        width=b;
    }
    double area() {
        System.out.print("Area is ");
        return length*width;
    }
}
class Box extends Rectangle{
    double depth;
    Box(){

    }
    Box(double x,double y,double z){
        length=x;
        width=y;
        depth=z;
    }
    double area() {
        System.out.print("Area is ");
        return (2*length*width+2*length*depth+2*width*depth);
    }
    double volume() {
        System.out.print("Volume is ");
        return length*width*depth;
    }
}
```

```java
public class Main {

    public static void main(String[] args) {
        Rectangle r1=new Rectangle(4,6);
        System.out.println(r1.area());

        Box b1=new Box(5,8,2);
        System.out.println(b1.area());
        System.out.println(b1.volume());
    }

}
```



2. Write a Java program to create a class known as "BankAccount"
    - with attributes
        - bank name
        - account number
        - balance amount
    - and methods
        - deposit()
        - withdraw().
        - getBalance()
        - setBalance()

    Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below five hundred.

    Write a Main class with main() function. Create the object of SavingsAccount and call the deposit() and withdraw() functions to test the minimum balance condition.

```java
package Questions;
class BankAccount{
    String BN;
    int accno;
    double accbal;
    BankAccount(){

    }
    BankAccount(String a,int b, double c){
        BN=a;
        accno=b;
        accbal=c;
    }
    void deposit(double x) {
        accbal+=x;
        System.out.println("Amount deposited --- Account Balance is");
        System.out.println(accbal);
    }
    void withdraw(double x) {
        accbal-=x;
        System.out.println("Amount withdrawn");
    }
    double getBalance() {
        System.out.print("Account Balance is");
        return accbal;
    }
    void setBalance(double x) {
        accbal=x;
    }
}
```

```java
class SavingsAccount extends BankAccount{
    SavingsAccount(String x,int y,double z){
        BN=x;
        accno=y;
        accbal=z;
    }
    void withdraw(double x) {
        if(accbal<500) {
            System.out.println("Cannot withdraw --- Insufficient Funds");
        }
        else {
            accbal-=x;
            System.out.println("Amount withdrawn --- Account Balance is");
            System.out.println(accbal);
        }
    }
}
public class Mains {

    public static void main(String[] args) {
        SavingsAccount s1=new SavingsAccount("ICICI",12,2500);
        s1.deposit(500);
        s1.withdraw(2600);
        s1.withdraw(500);
    }

}
```

```
Amount deposited --- Account Balance is
3000.0
Amount withdrawn --- Account Balance is
400.0
Cannot withdraw --- Insufficient Funds
```

3. Write a Java program to create a class known as Person
    - With attributes
        - First name
        - Last name
    - with methods
        - constructors – default, parameterized
        - getFirstName()
        - getLastName().

    Create a subclass called Employee
    - with attribute
        - employee ID
        - job title
    - and methods
        - constructors – default, parameterized
        - getEmployeeId()
        - overrides the getLastName() method to include the employee's job title.

Write a Main class with main() function. Create two objects of the Employee class in the main(). One object with default constructor and the second one with parameterized constructor. The parameters should be passed from the Employee constructor to the Person constructor.

```java
package Questions;
class Person{
    String fn,ln;
    Person(){
        fn="Bhadra";
        ln="R";
    }
    Person(String a,String b){
        fn=a;
        ln=b;
    }
    void getFirstName() {
        System.out.println("First name is "+fn);
    }
    void getLastName() {
        System.out.println("Last name is "+ln);
    }
}
class Employee extends Person{
    String JT;
    int EMPID;
    Employee(){
        super();
        EMPID=6;
        JT="Doctor";
    }
    Employee(String m,String n,int p,String q){
        super(m,n);
        EMPID=p;
        JT=q;
    }
    void getEmployeeid(){
        System.out.println("Employee ID is "+EMPID);
    }
    void getLastName() {
        System.out.println("Last name is "+ln+" and job title is "+JT);
    }
}
public class MAlN {

    public static void main(String[] args) {
        Employee e1=new Employee();
        Employee e2=new Employee("Devi","Karthika",3,"Gynaecologist");
        e1.getFirstName();
        e1.getLastName();
        e2.getEmployeeid();
    }

}
```

```
First name is Bhadra
Last name is R and job title is Doctor
Employee ID is 3
```