OOPJ
Assignment-5

1. Create a base class BankAccount with methods like deposit() and withdraw(). Derive a class SavingsAccount that overrides the withdraw() method to impose a limit on the withdrawal amount. Write a program that demonstrates the use of overridden methods and proper access modifiers & return the details.

```
package com.example.a5q1;

import java.util.Scanner;

class BankAccount {

	private long accountNumber;
	private double balance;

	public BankAccount() {
		this(12345678, 100000.0);
	}

	public BankAccount(long accountNumber, double balance) {
		this.accountNumber = accountNumber;
		this.balance = balance;
	}

	public void deposit(double amount) {
		if( amount > 0 ) {
			balance += amount;
			System.out.printf("Deposited: %.2f \n", amount);
			System.out.printf("Balance: %.2f \n\n", this.getBalance());
		} else {
			System.out.println("Deposit amount must be positive");
		}
	}

	public void withdraw(double amount) {
		if( balance > amount ) {
			balance -= amount;
			System.out.printf("Withdrew: %.2f \n", amount);
			System.out.printf("Balance: %.2f \n\n", this.getBalance());
		} else {
			System.out.println("Insufficient balance");
		}
	}

	public long getAccountNumber() {
		return accountNumber;
	}

	public double getBalance() {
		return balance;
```

```
        }

        public void setBalance(double balance) {
                this.balance = balance;
        }
}

class SavingsAccount extends BankAccount {

        public void withdraw(double amount) {
                if( this.getBalance() > amount ) {
                        if(amount > 20000)
                                System.out.println("You can withdraw 20000 at a time.");
                        else if(this.getBalance() >= amount) {
                                this.setBalance(this.getBalance() - amount);
                                System.out.printf("Withdrew: %.2f \n", amount);
                                System.out.printf("Balance: %.2f \n\n", this.getBalance());
                        } else
                                System.out.println("Insufficient balance");
                }
        }
}

public class Program {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);

                SavingsAccount sa = new SavingsAccount();

                sa.withdraw(15000);
                sa.deposit(10000);
                sa.deposit(25000);
                sa.withdraw(30000);
        }
}
```

Output

```
Withdrew: 15000.00
Balance: 85000.00

Deposited: 10000.00
Balance: 95000.00

Deposited: 25000.00
Balance: 120000.00

You can withdraw 20000 at a time.
```

2) Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.

```java
package com.example.a5q2;

class Vehicle {

	private String make;
	private int year;

	public Vehicle() {
		this.make = "";
		this.year = 0;
	}
	public Vehicle(String make, int year) {
		this.make = make;
		this.year = year;
	}

	public String getMake() {
		return make;
	}

	public void setMake(String make) {
		this.make = make;
	}

	public int getYear() {
		return year;
	}

	public void setYear(int year) {
		this.year = year;
	}

	public void displayRecord() {
		System.out.println("Make: " + this.getMake());
		System.out.println("Year: " + this.getYear());
	}
}
class Car extends Vehicle {

	private String model;
	public Car() {
		this.model = "";
	}
	public Car(String make, int year, String model) {
		super(make, year);
		this.model = model;
	}
	public String getModel() {
		return model;
	}
	public void setModel(String model) {
		this.model = model;
	}

	@Override
	public void displayRecord() {
```

```
                super.displayRecord();
                System.out.println("Model: " + this.getModel());
        }
}
public class Program {
        public static void main(String[] args) {

                Vehicle v = new Vehicle();
                Car c = new Car("Honda", 2023, "Creta");

                c.displayRecord();
        }
}
```

Output

```
Make: Honda
Year: 2023
Model: Creta
```

3. Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.

```
package com.example.a5q3;

class Animal {
        private String name;

        public void eat() {
                System.out.println("I can eat");
        }

        public void sleep() {
                System.out.println("I can sleep");
        }
}

class Dog extends Animal {
        public void bark() {
                System.out.println("I can bark");
        }
}

public class Program {
        public static void main(String[] args) {
                Animal animal = new Animal();
                animal.eat();
                animal.sleep();
```

```
                System.out.println();

                Dog dog = new Dog();
                dog.eat();
                dog.sleep();
                dog.bark();
        }
}
```

Output

```
I can eat
I can sleep

I can eat
I can sleep
I can bark
```

4. Build a class Student which contains details about the Student and compile and run its instance.

```
package com.assignment.demo;

import java.util.Scanner;

class Student {
        private String name;
        private int age;
        private int rollNo;
        private String address;

        public Student() {
                this("", 0, 0, "");
        }

        public Student(String name, int age, int rollNo, String address) {
                this.name = name;
                this.age = age;
                this.rollNo = rollNo;
                this.address = address;
        }

        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public int getAge() {
                return age;
        }
        public void setAge(int age) {
                this.age = age;
```

```
	}
	public int getRollNo() {
		return rollNo;
	}
	public void setRollNo(int rollNo) {
		this.rollNo = rollNo;
	}
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
	}
}

class StudentUtil {

	private Student std = new Student();
	public Student getStd() {
		return std;
	}

	private static Scanner sc = new Scanner(System.in);

	public void acceptRecord() {
		System.out.print("Enter name: ");
		this.std.setName(sc.nextLine());
		System.out.print("Enter age: ");
		this.std.setAge( sc.nextInt() );
		System.out.print("Enter Roll No: ");
		this.std.setRollNo( sc.nextInt() );
		sc.nextLine();
		System.out.print("Enter Address: ");
		this.std.setAddress( sc.nextLine() );
	}

	public void printRecord() {
		System.out.println("Name: " + std.getName());
		System.out.println("Age: " + std.getAge());
		System.out.println("Roll No: " + std.getRollNo());
		System.out.println("Address: " + std.getAddress());
	}
}
public class Program {
	public static void main(String[] args) {

		StudentUtil util = new StudentUtil();

		util.acceptRecord();
		util.printRecord();
	}
}
```

Output

```
Enter name: Chitransh
Enter age: 27
Enter Roll No: 30
Enter Address: Raebareli
Name: Chitransh
Age: 27
Roll No: 30
Address: Raebareli
```

5) Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.

```
package com.assignment.demo;

class Vehicle {
	public void startEngine() {

	}

	public void stopEngine() {

	}
}
class Car extends Vehicle {
	@Override
	public void startEngine() {
		System.out.println("Car engine started");
	}

	@Override
	public void stopEngine() {
		System.out.println("Car engine stopped");
	}
}
class Motorcycle extends Vehicle {
	@Override
	public void startEngine() {
		System.out.println("Motorcycle engine started");
	}

	@Override
	public void stopEngine() {
		System.out.println("Motorcycle engine stopped");
	}
}
public class Program {
	public static void main(String[] args) {

		Car c = new Car();
		c.startEngine();
		c.stopEngine();
		System.out.println();

		Motorcycle mc = new Motorcycle();
```

```
                mc.startEngine();
                mc.stopEngine();
        }
}
```

Output

```
Car engine started
Car engine stopped

Motorcycle engine started
Motorcycle engine stopped
```