

SCHOOL OF
COMPUTING

Ganath Avinash
CH.SC.U4CSE24118
OBJECT ORIENTED PROGRAMMING
(23CSE111)
LAB RECORD



**SCHOOL OF
COMPUTING**

**AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF COMPUTING, CHENNAI**

BONAFIDE CERTIFICATE

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by **CH.SC.U4CSE24118 – Ganath Avinash G.R** in “**Computer Science and Engineering**” is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on

Internal Examiner 1

Internal Examiner 2

INDEX

S.NO	TITLE	PAGE.NO
UML DIAGRAM		
1.	ATM Model type of Application	
	1.a) Use Case Diagram	4
	1.b) Class Diagram	5
	1.c) Sequence Diagram	5
	1.d) Collab Diagram	6
	1.e) State Diagram	6
2.	Hospital Management Application	
	2.a) Use Case Diagram	7
	2.b) Class Diagram	7
	2.c) Object Diagram	8
	2.d) State Diagram	8
	2.e) Sequence Diagram	9
3.	Basic Java Programs	
	3.a) Even Or Odd	10
	3.b) Count Number Of Digits	11
	3.c) Factorial	12
	3.d) Fibonacci Series	13
	3.e) Largest Number Calculator	14
	3.f) Multiplication Table	15
	3.g) Prime Check	16
	3.h) Reverse Number	17
	3.i) Sum of N Natural Numbers	18
	3.j) Sum of Digits	19
4.	INHERITANCE	
	SINGLE INHERITANCE PROGRAMS	

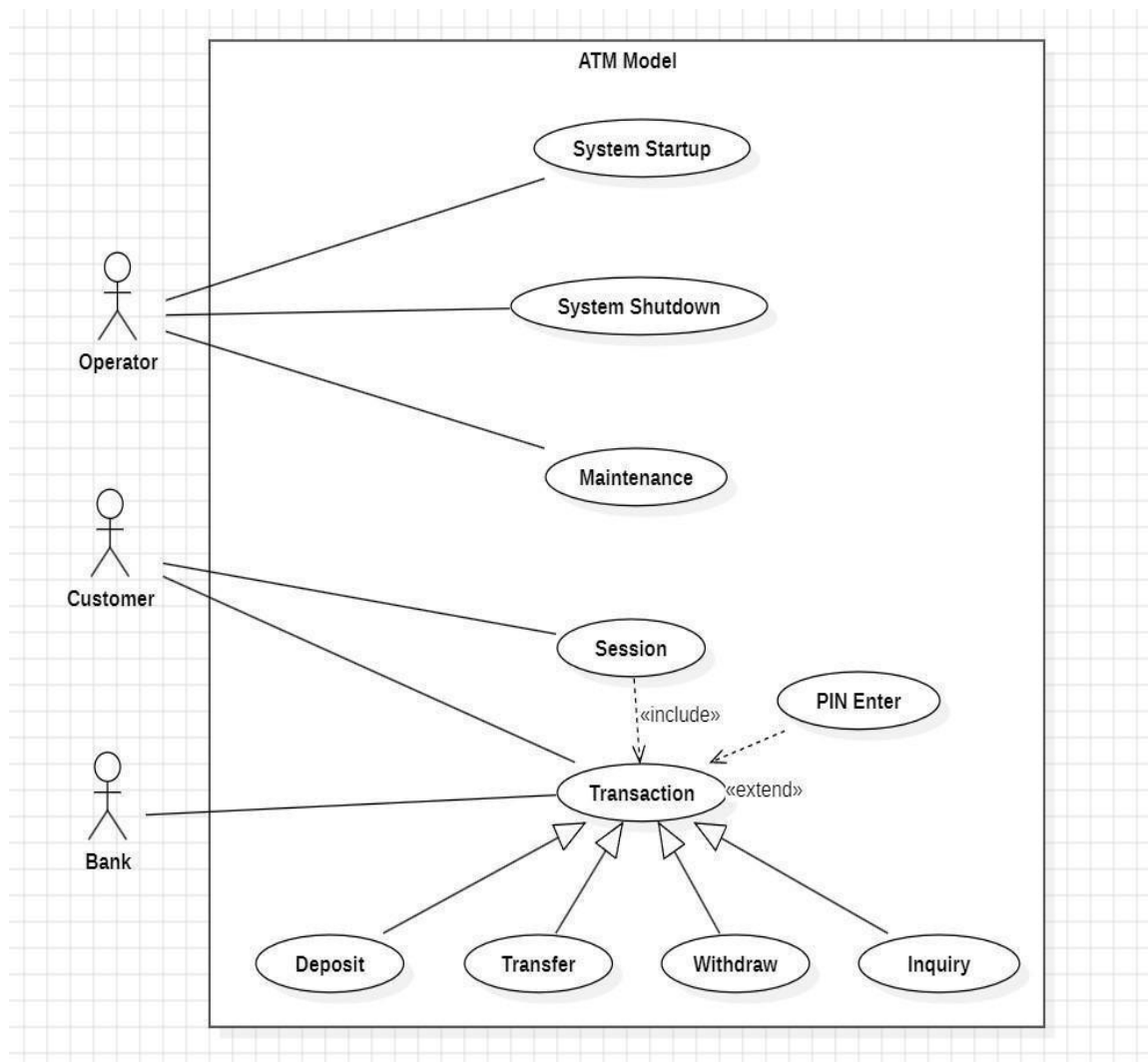
	4.a)Single Inheritance 1	20
	4.b) Single Inheritance 2	21
5.	MULTIPLE INHERITANCE PROGRAMS	
	5.a)Multiple Inheritance 1	22
	5.b) Multiple Inheritance 2	23
6.	HIERACHICAL INHERITANCE PROGRAMS	
	6.a)Hierachial Inheritance 1	24
	6.b) Hierachial Inheritance 2	25
7.	HYBRID INHERITANCE PROGRAMS	
	7.a)Hybrid Inheritance 1	26
	7.b) Hybrid Inheritance 2	28
	POLYMORPHISM	
8.	CONSTRUCTOR PROGRAMS	
	8.a)Constructor Book	29
9.	CONSTRUCTOR OVERLOADING PROGRAMS	
	9.a)Multiple Constructor Book	30
10.	METHOD OVERLOADING PROGRAMS	
	10.a)Calclator Program	31
11.	METHOD OVERRIDING PROGRAMS	
	11.a)Overriding 1	32
	11.b) Overriding 2	33
	ABSTRACTION	
12.	ABSTRACT CLASS PROGRAMS	
	12.a)Abstract Program 1	34
	12.b) Abstract Program 2	36
	12.c) Abstract Program 3	37

	12.d) Abstract Program 4	38
13.	INTERFACE PROGRAMS	
	13.a)Interface Problem 1	39
	13.b) Interface Problem 2	40
	13.c) Interface Problem 3	41
	13.d) Interface Problem 4	42
	ENCAPSULATION	
14.	ENCAPSULATION PROGRAMS	
	14.a)Encapsulaion program 1	43
	14.b) Encapsulaion program 2	44
	14.c) Encapsulaion program 3	45
	14.d) Encapsulaion program 4	46
	14.e) Encapsulaion program 5	47
	14.f) Encapsulaion program 6	48
	14.g) Encapsulaion program 7	49
	14.h) Encapsulaion program 8	50
15.	PACKAGES PROGRAMS	
	15.a)Built in 1,2,3	51
	15.b)Built in 4,5,6	51
	15.c)User Defined 1	52
	15.d)User Defined 2	53
16.	EXCEPTION HANDLING PROGRAMS	
	16.a)Exception Handling 1	56
	16.b) Exception Handling 2	57
	16.c) Exception Handling 3	58
	16.d) Exception Handling 4	59
17.	FILE HANDLING PROGRAMS	
	17.a)Writing Using FileWriter	60
	17.b) Writing Using BufferedWriter	60
	17.c)Reading Using FileReader	61
	17.d) Reading Using BufferedReader	62

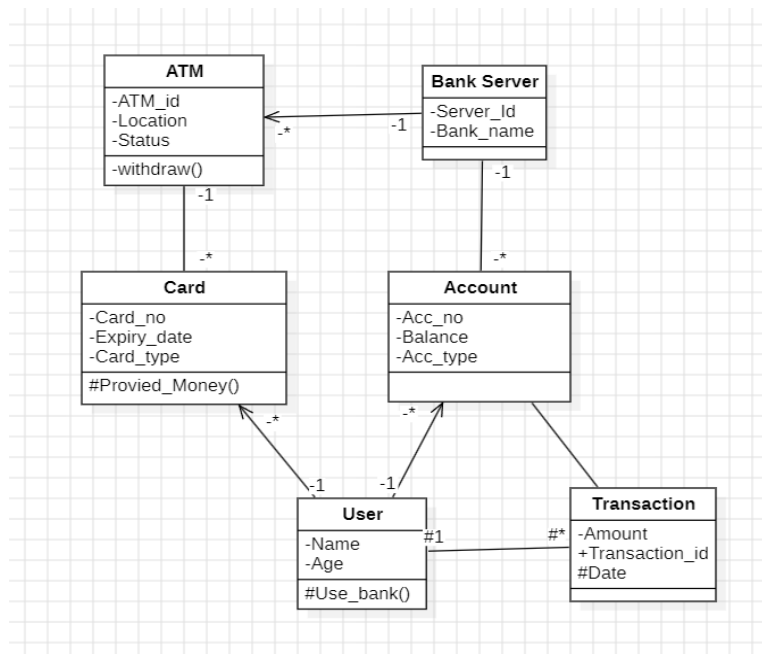
UML DIAGRAMS

1. ATM Model Application

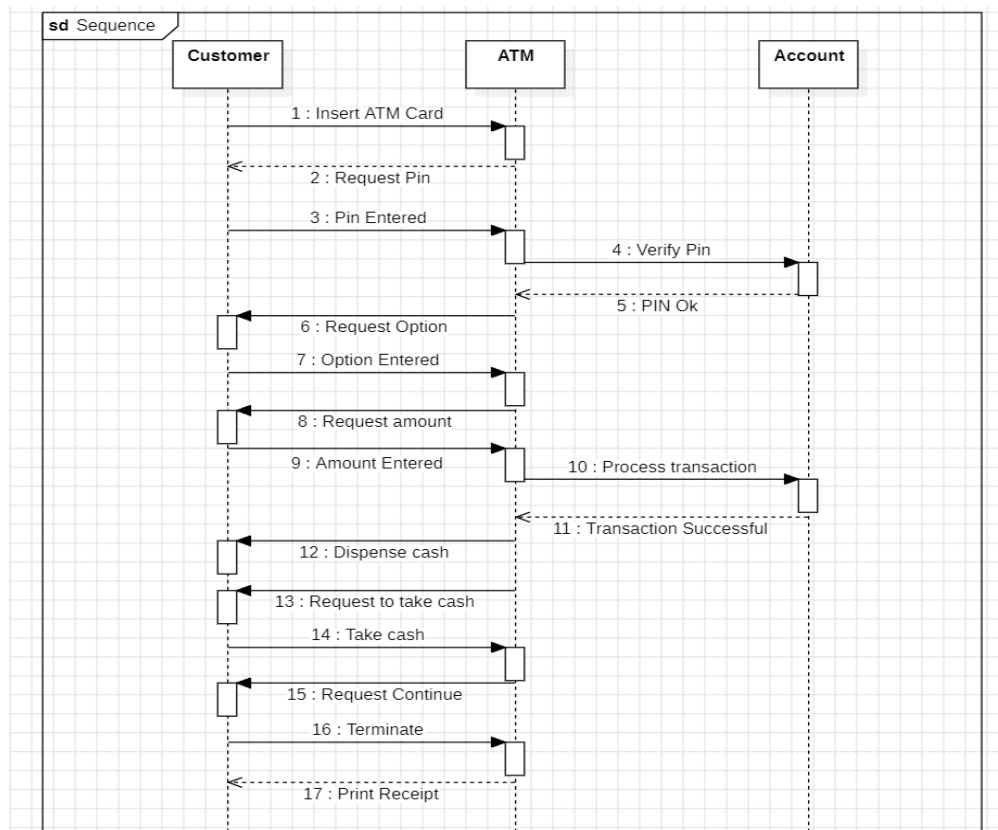
1.a) Use Case Diagram:



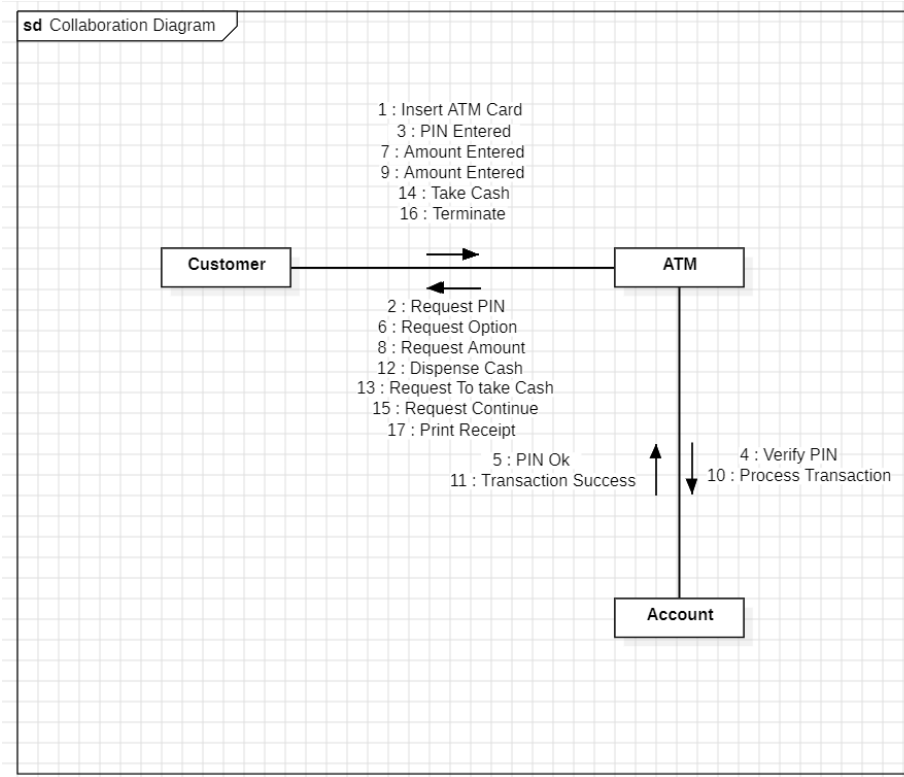
1.b) Class Diagram:



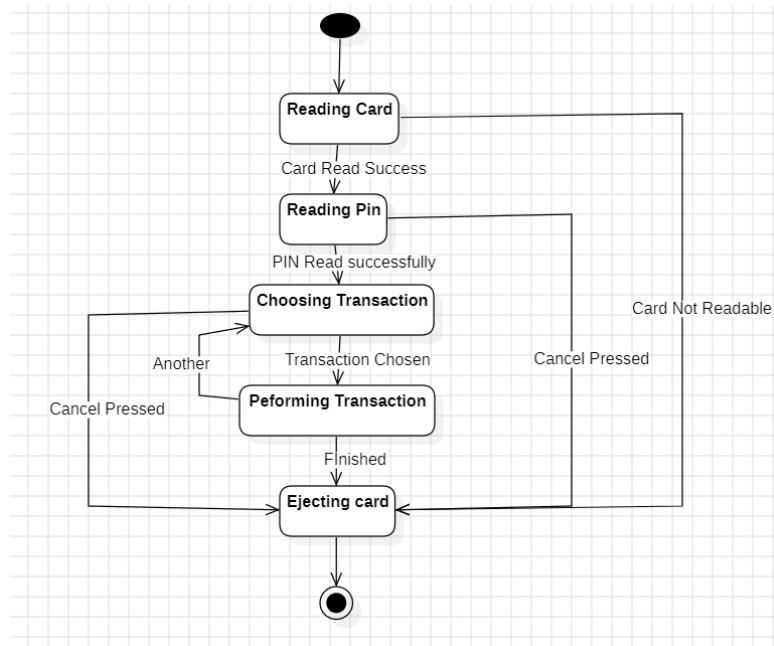
1.c) Sequence Diagram:



1.d) Collab Diagram:

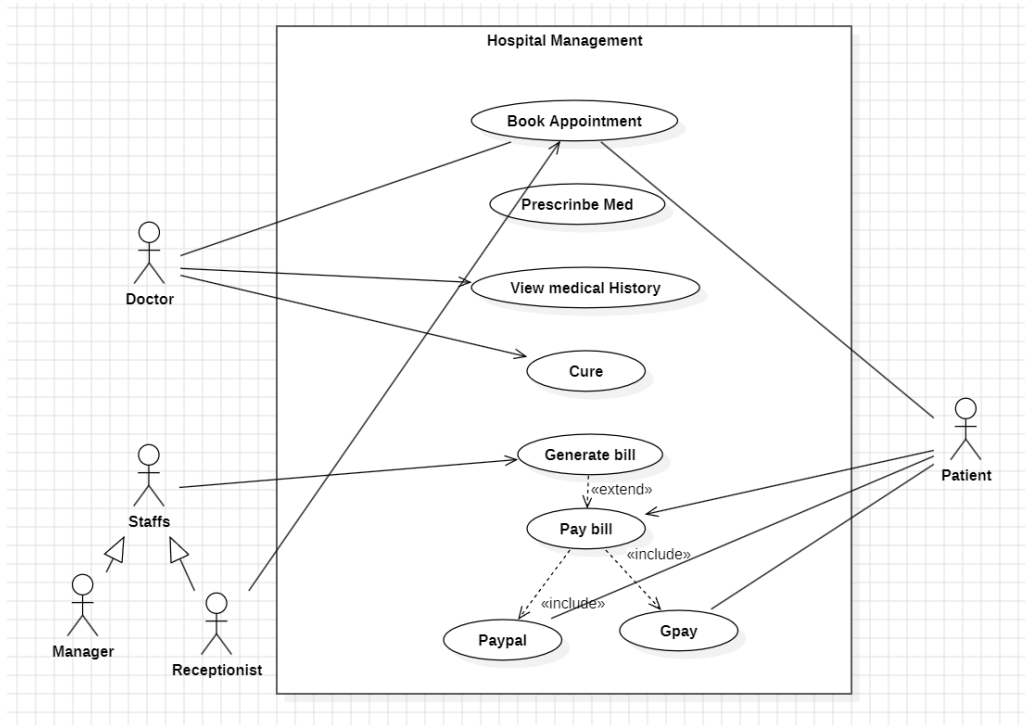


1.e) State Diagram:

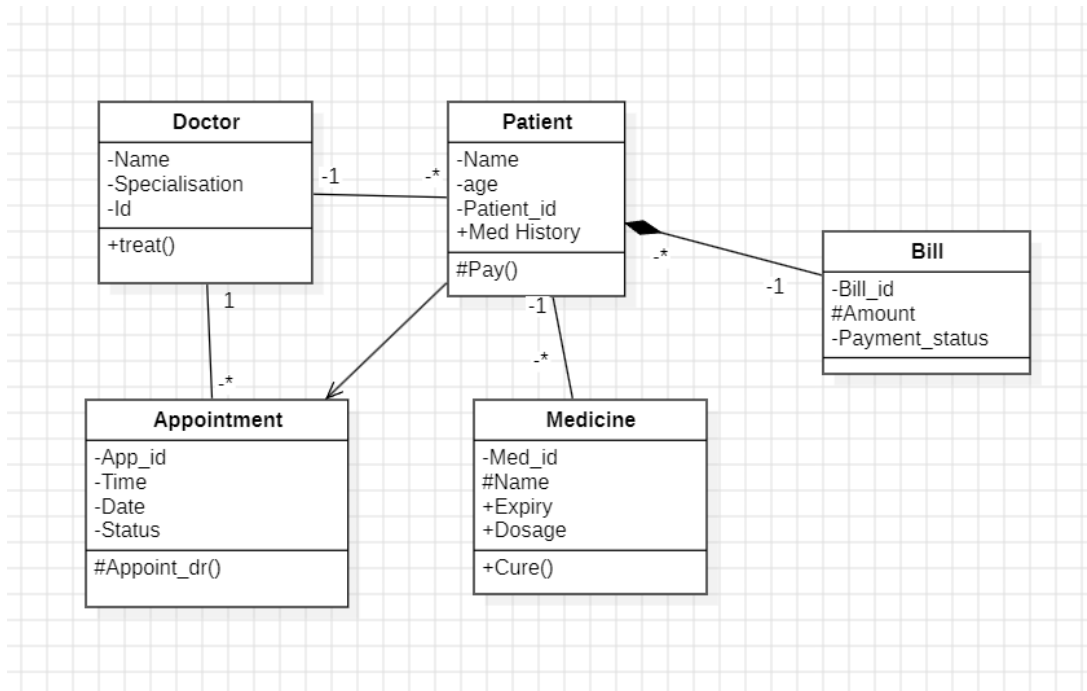


2. Hospital Management

2.a) Use Case Diagram:

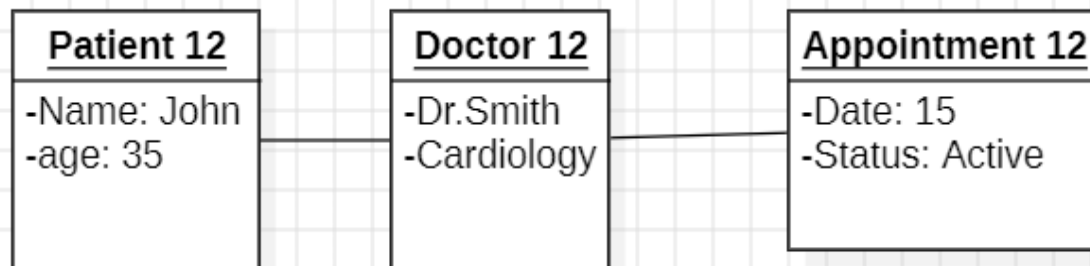


2.b) Class Diagram:



2c) Object Diagram:

Object Diagram

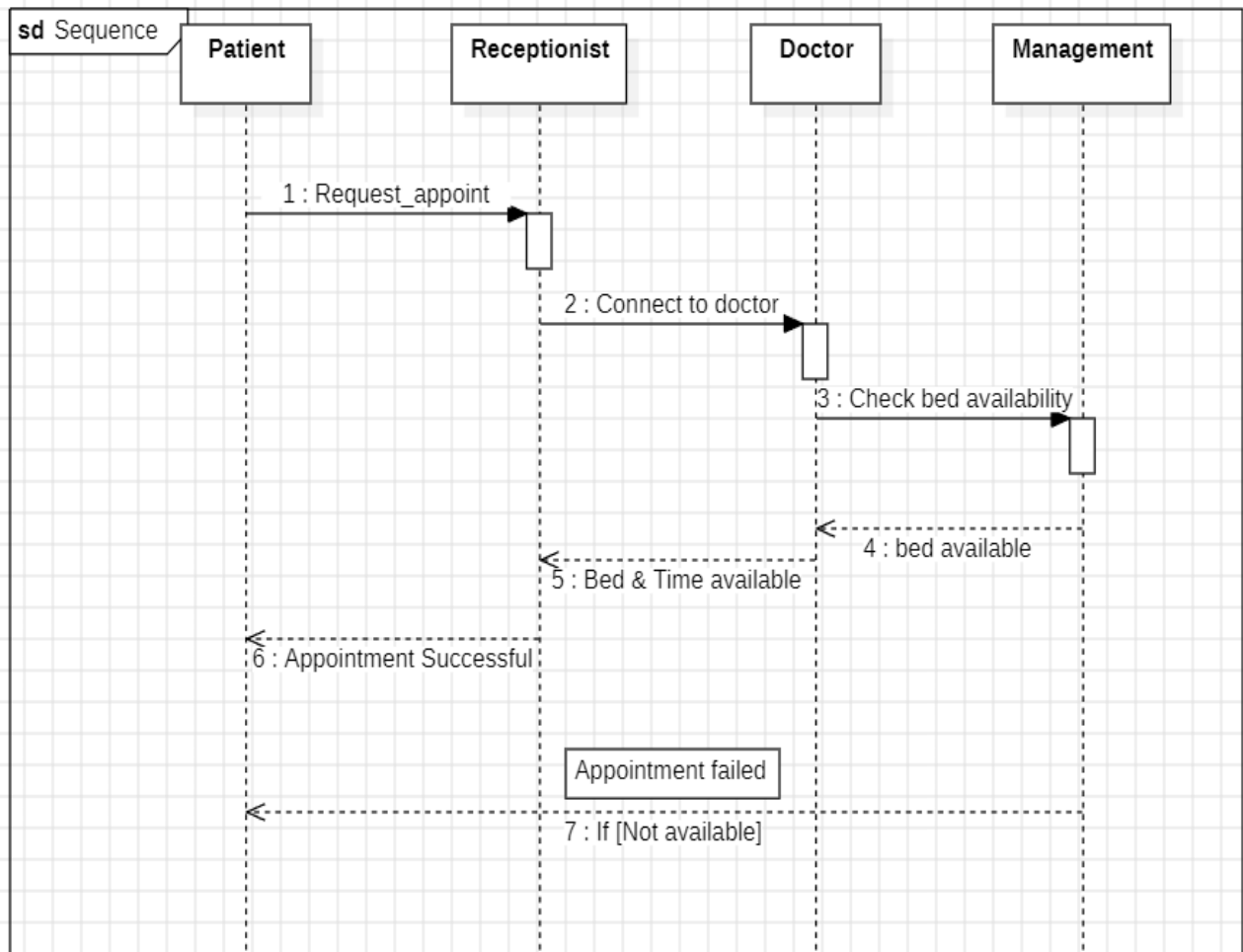


2d) State Diagram:

State Diagram



2e)Sequence Diagram:



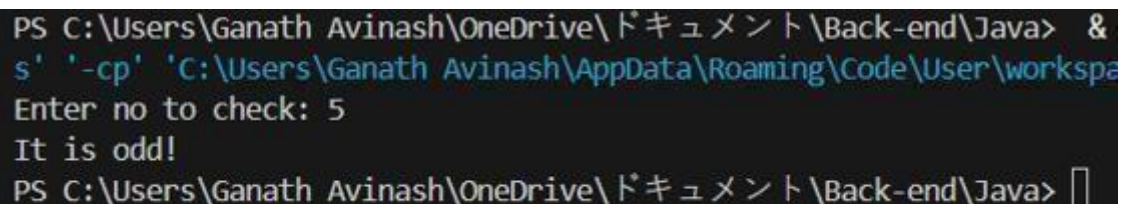
Basic Java Questions

3a) Even Or Odd with Scanner:

Code:

```
import java.util.Scanner;
public class even{
    public void find(int a){
        if(a>=0){
            if (a%2==0){
                System.out.println("It is even!");
            }
            else{
                System.out.println("It is odd!");
            }
        }
        else{
            System.out.println("Enter a number greater than or equal to 0!!");
        }
    }
    public static void main(String[]args){
        Scanner ip = new Scanner(System.in);
        even ob1=new even();
        System.out.print("Enter no to check: ");
        int a=ip.nextInt();
        ob1.find(a);
    }
}
```

Output:



```
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java> &
s' '-cp' 'C:\Users\Ganath Avinash\AppData\Roaming\Code\User\workspa
Enter no to check: 5
It is odd!
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java> █
```

3b) Count Number Of Digits :

Code:

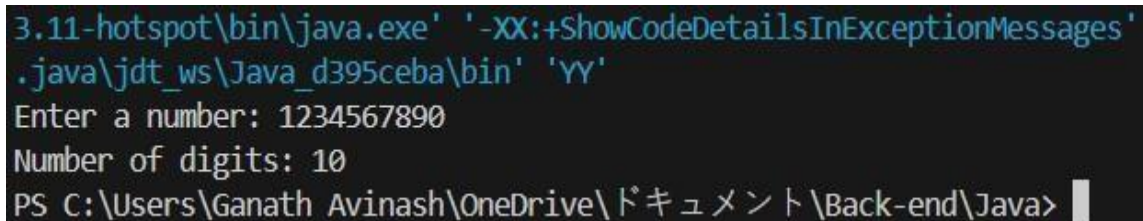
```
import java.util.Scanner;

public class CountDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        int count = 0;
        while (num != 0) {
            num /= 10;
            count++;
        }

        System.out.println("Number of digits: " + count);
        scanner.close();
    }
}
```

Output:



```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter a number: 1234567890
Number of digits: 10
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

3c) Factorial:

Code:

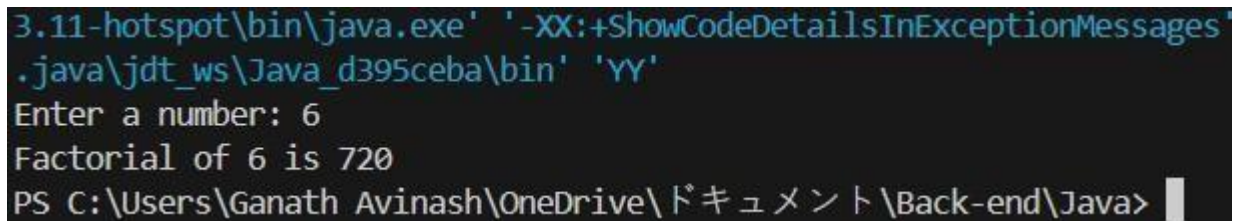
```
import java.util.Scanner;

public class FactorialLoop {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        int factorial = 1;
        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of " + num + " is " + factorial);
        scanner.close();
    }
}
```

Output:



The screenshot shows a terminal window with the following text:

```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter a number: 6
Factorial of 6 is 720
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

3d) Fibonacci Series:

Code:

Code:

```
import java.util.Scanner;

public class FactorialLoop {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        int factorial = 1;
        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of " + num + " is " + factorial);
        scanner.close();
    }
}
```

Output:

```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter the number of terms: 8
Fibonacci Series: 0 1 1 2 3 5 8 13
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java> █
```

3e) Largest Number Calculator:

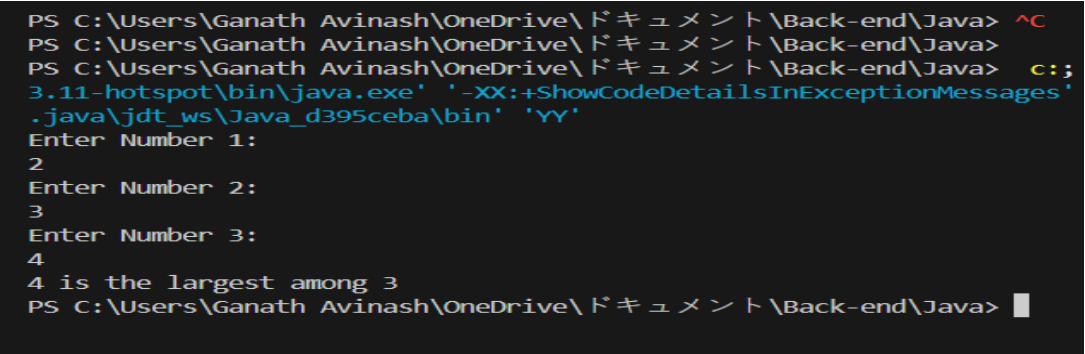
Code:

```
import java.util.Scanner;

public class Largest{
    int a,b,c;
    void lar(int a,int b, int c){
        if(a>b && a>c){
            System.out.println(a + "Is the largest among 3");
        }
        else if(b>a && b>c){
            System.out.println(b + "Is the largest among 3");
        }
        else if(c>a && c>b){
            System.out.println(c + "Is the largest among 3");
        }
        else{
            System.out.println("All are equal no larger number");
        }
    }
}

class call{
    public static void main(String[]args){
        Largest l1= new Largest();
        Scanner ip=new Scanner(System.in);
        System.out.println("Enter Number 1: ");
        int a=ip.nextInt();
        System.out.println("Enter Number 2: ");
        int b=ip.nextInt();
        System.out.println("Enter Number 3: ");
        int c=ip.nextInt();
        l1.lar(a,b,c);
    }
}
```

Output:



```
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java> ^C
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java> c:;
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'yy'
Enter Number 1:
2
Enter Number 2:
3
Enter Number 3:
4
4 is the largest among 3
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java> █
```


3f) Multiplication Table :

Code:

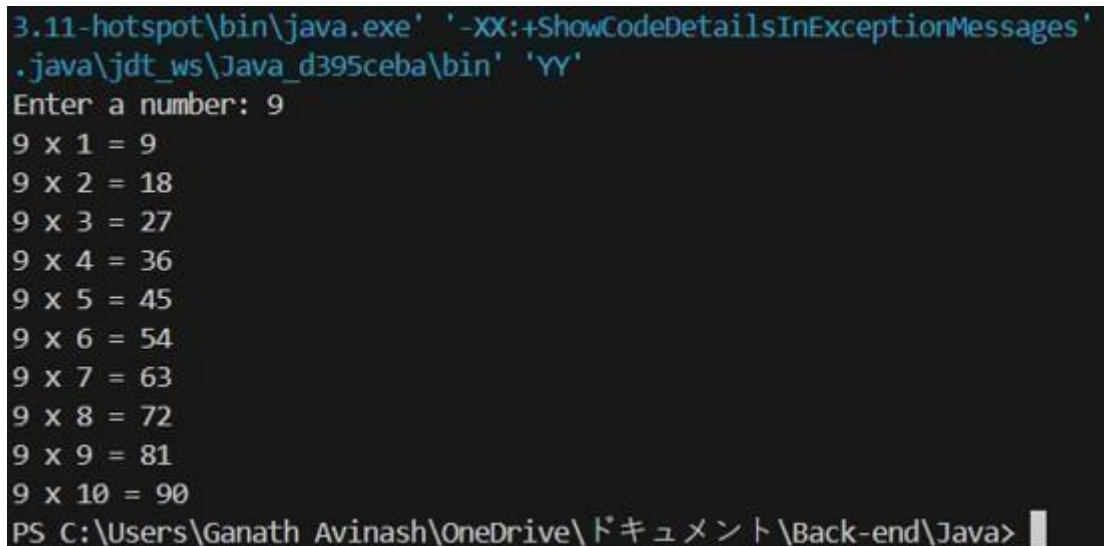
```
import java.util.Scanner;

public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        for (int i = 1; i <= 10; i++) {
            System.out.println(num + " x " + i + " = " + (num * i));
        }

        scanner.close();
    }
}
```

Output:



```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter a number: 9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

3g) Prime Check:

Code:

```
import java.util.Scanner;

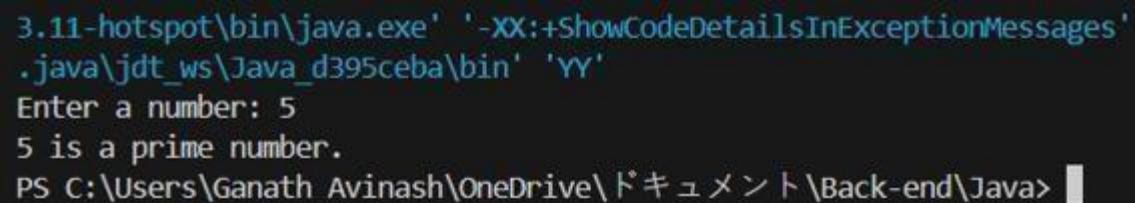
public class PrimeCheck {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();
        boolean isPrime = true;

        if (num <= 1) {
            isPrime = false;
        } else {
            for (int i = 2; i <= num / 2; i++) {
                if (num % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

        if (isPrime)
            System.out.println(num + " is a prime number.");
        else
            System.out.println(num + " is not a prime number.");

        scanner.close();
    }
}
```

Output:



```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter a number: 5
5 is a prime number.
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

3h) Reverse Number:

Code:

```
import java.util.Scanner;

public class ReverseNumber {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

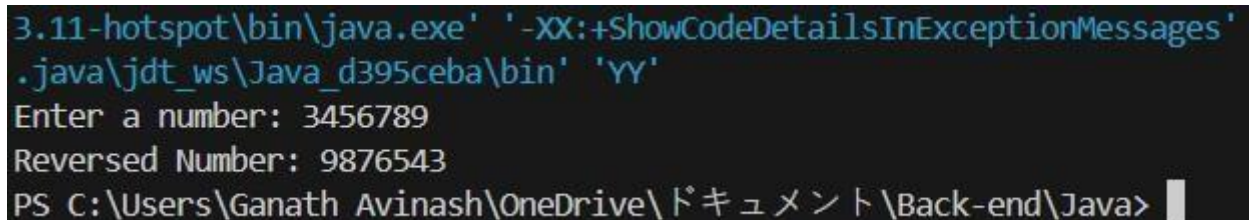
        System.out.print("Enter a number: ");

        int num = scanner.nextInt();

        int reversed = 0;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }

        System.out.println("Reversed Number: " + reversed);
        scanner.close();
    }
}
```

Output:



```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter a number: 3456789
Reversed Number: 9876543
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

3i) Sum Of N Natural Numbers:

Code:

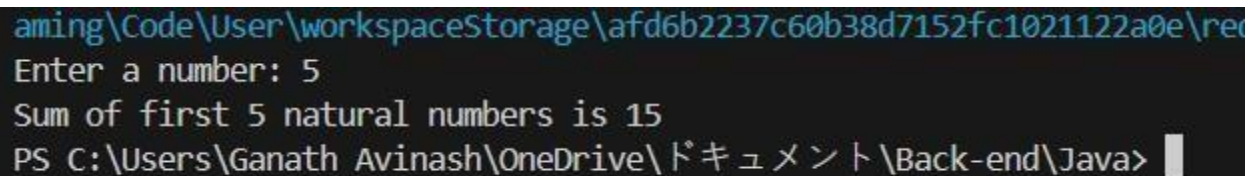
```
import java.util.Scanner;

public class SumNaturalNumbers {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = scanner.nextInt();

        int sum = 0, i = 1;
        while (i <= n) {
            sum += i;
            i++;
        }

        System.out.println("Sum of first " + n + " natural numbers is " +
sum);
        scanner.close();
    }
}
```

Output:



```
aming\Code\User\workspaceStorage\afd6b2237c60b38d7152fc1021122a0e\re
Enter a number: 5
Sum of first 5 natural numbers is 15
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

3j) Sum of Digits:

Code:

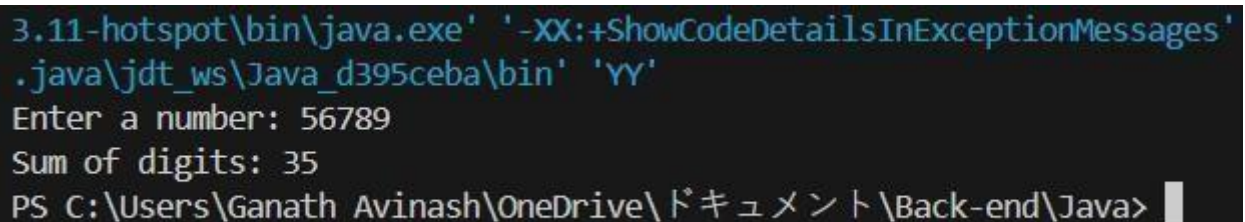
```
import java.util.Scanner;

public class SumOfDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = scanner.nextInt();

        int sum = 0;
        while (num != 0) {
            sum += num % 10;
            num /= 10;
        }

        System.out.println("Sum of digits: " + sum);
        scanner.close();
    }
}
```

Output:



```
3.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
.java\jdt_ws\Java_d395ceba\bin' 'YY'
Enter a number: 56789
Sum of digits: 35
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java>
```

EXP: 4,5,6,7	INHERITANCE	OOPS
--------------	-------------	------

4a)

Code:

```

public class I1{
    public static void main(String[]args){
        Car c1=new BMW();
        BMW b1=new BMW();
        Car c2=new Car();
        c1.stt();
        b1.stt();
        b1.drive();
    }
}

class Car{
    void stt(){
        System.out.println("Car Starts");
    }
}

class BMW extends Car{
    void drive(){
        System.out.println("BMW Is Driven");
    }
}

```

Screenshot:

```

C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>javac I1.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I1.java
Car Starts
Car Starts
BMW Is Driven
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>

```

4b)

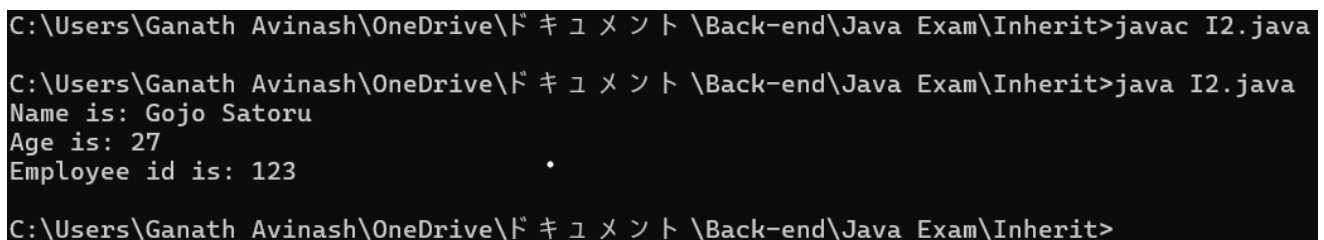
Code:

```
public class I2{
public static void main (String[]args){
emp e1=new emp(123,"Gojo Satoru",27);
e1.di();
e1.emdi();
}
}

class Person{
int age;
String name;
Person(String name,int age){
this.name=name;
this.age=age;
}
void di(){
System.out.println("Name is: "+name);
System.out.println("Age is: "+age);
}
}

class emp extends Person{
int emid;
emp(int emid,String name,int age){
super(name,age);
this.emid=emid;
}
void emdi(){
System.out.println("Employee id is: "+emid);
}
}
```

Screenshot:



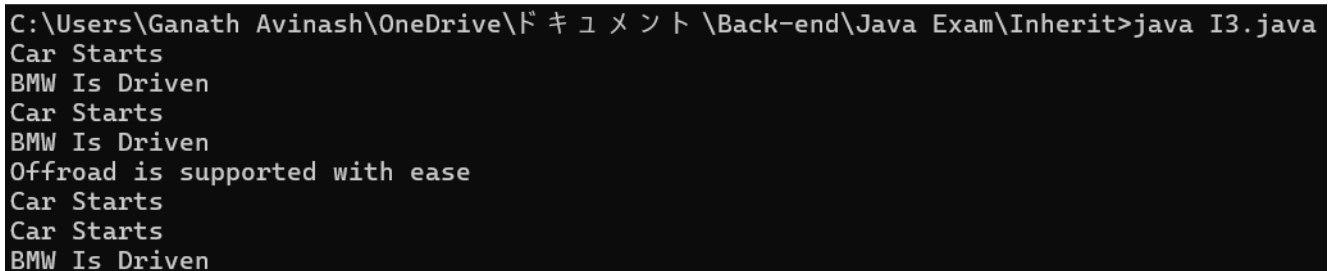
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>javac I2.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I2.java
Name is: Gojo Satoru
Age is: 27
Employee id is: 123
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>
```

5a)

Code:

```
public class I3{
public static void main(String[]args){
Car c1=new BMW();
BMW b1=new BMW();
Car c2=new Car();
BMW t3=new BMWTruck();
BMWTruck t4=new BMWTruck();
t3.stt();
t3.drive();
t4.stt();
t4.drive();
t4.offroad();
c1.stt();
b1.stt();
b1.drive();
}
}
class Car{
void stt(){
System.out.println("Car Starts");
}
}
class BMW extends Car{
void drive(){
System.out.println("BMW Is Driven");
}
}
class BMWTruck extends BMW{
void offroad(){
System.out.println("Offroad is supported with ease");
}}
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I3.java
Car Starts
BMW Is Driven
Car Starts
BMW Is Driven
Offroad is supported with ease
Car Starts
Car Starts
BMW Is Driven
```

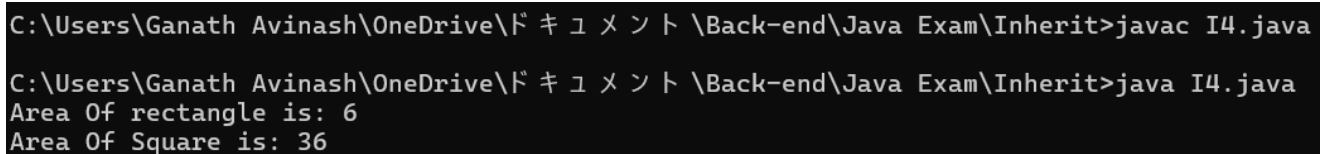

5b)

Code:

```
public class I4{
public static void main(String[]args){
rect r1=new rect();
r1.area(2,3);
square s1=new square();
s1.area(6);
}
}

class Shape{
void area(){
System.out.println("This can Calculate area of rectangle and square");
}
}
class rect extends Shape{
void area(int a,int b){
System.out.println("Area Of rectangle is: "+(a*b));
}
}
class square extends rect{
void area(int a){
System.out.println("Area Of Square is: "+(a*a));
}
}
```

Screenshot:



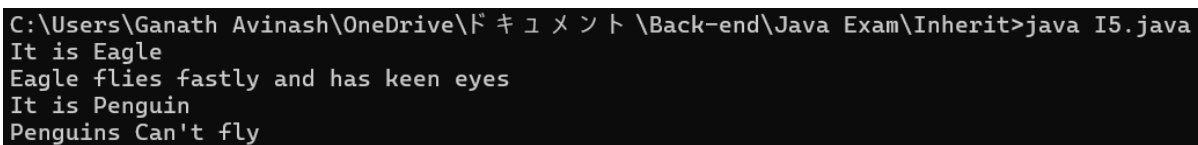
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>javac I4.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I4.java
Area Of rectangle is: 6
Area Of Square is: 36
```

6a)

Code:

```
public class I5{
public static void main(String[]args){
eagle e1=new eagle("Eagle");
pen p1=new pen("Penguin");
e1.dis();
e1.fly();
p1.dis();
p1.fly();
}
}
class Bird{
String type;
void fly(){
System.out.println("Birds fly in various speeds");
}
void dis(){
System.out.println("It is "+type);
}
}
class eagle extends Bird{
eagle(String typ){
super.type=typ;
}
void fly(){
System.out.println("Eagle flies fastly and has keen eyes");
}
}
class pen extends Bird{
pen(String typ){
super.type=typ;
}
void fly(){
System.out.println("Penguins Can't fly");
}
}
```

Screenshot:



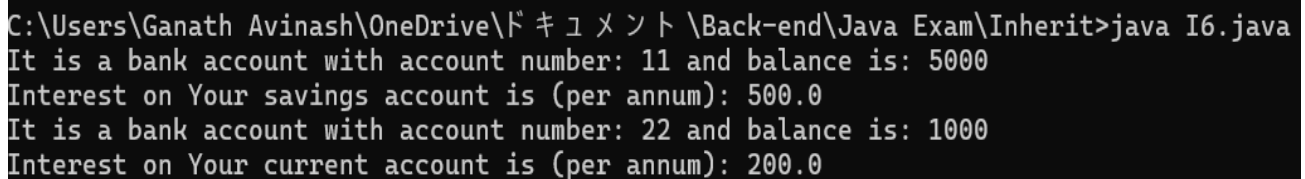
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I5.java
It is Eagle
Eagle flies fastly and has keen eyes
It is Penguin
Penguins Can't fly
```

6b)

Code:

```
public class I6{
public static void main(String[]args){
sacc s1=new sacc(11,5000);
cacc c1=new cacc(22,1000);
s1.bb();
s1.calci();
c1.bb();
c1.calca();}}
class BankAcc{
int accno;
int balance;
BankAcc(int accno,int balance){
this.accno=accno;
this.balance=balance;}
void bb(){
System.out.println("It is a bank account with account number: "+accno+" and balance is:
"+balance);}}
class sacc extends BankAcc{
sacc(int accno,int balance){
super(accno,balance);}
void calci(){
System.out.println("Interest on Your savings account is (per annum): "+balance*0.1);
}}
class cacc extends BankAcc{
cacc(int accno,int balance){
super(accno,balance);
}
void calca(){
System.out.println("Interest on Your current account is (per annum): "+balance*0.2);
}}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I6.java
It is a bank account with account number: 11 and balance is: 5000
Interest on Your savings account is (per annum): 500.0
It is a bank account with account number: 22 and balance is: 1000
Interest on Your current account is (per annum): 200.0
```

7a)

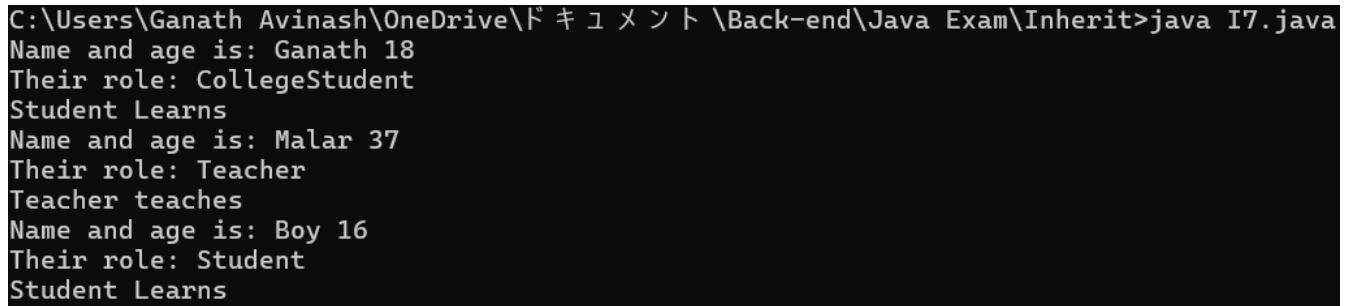
Code:

```
public class I7{
public static void main(String[]args){
cstudent c1=new cstudent("Ganath",18,"CollegeStudent");
teacher t1=new teacher("Malar",37,"Teacher");
student s1=new student("Boy",16,"Student");
c1.dis();
c1.learn();
t1.dis();
t1.teach();
s1.dis();
s1.learn();
}
}
```

```
class Person{
String name;
int age;
Person(String name,int age){
this.name=name;
this.age=age;
}
void dis(){
System.out.println("Name and age is: "+name+" "+age);
}
}
class teacher extends Person{
String role;
teacher(String name,int age,String role){
super(name,age);
this.role=role;
}
void teach(){
System.out.println("Their role: "+role);
System.out.println("Teacher teaches");
}
}
class student extends Person{
String role;
student(String name,int age,String role){
super(name,age);
this.role=role;
}
void learn(){
System.out.println("Their role: "+role);
```

```
System.out.println("Student Learns");
}
}
class cstudent extends student{
cstudent(String name,int age,String role){
super(name,age,role);
}
}
```

Screenshot:

A screenshot of a terminal window showing the execution of a Java program. The command prompt is 'C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I7.java'. The output consists of several lines of text: 'Name and age is: Ganath 18', 'Their role: CollegeStudent', 'Student Learns', 'Name and age is: Malar 37', 'Their role: Teacher', 'Teacher teaches', 'Name and age is: Boy 16', 'Their role: Student', and 'Student Learns'.

```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I7.java
Name and age is: Ganath 18
Their role: CollegeStudent
Student Learns
Name and age is: Malar 37
Their role: Teacher
Teacher teaches
Name and age is: Boy 16
Their role: Student
Student Learns
```

7b)

Code:

```
public class I8 {
    public static void main(String[] args) {
        // Hybrid inheritance demonstration
        SportsCar sportsCar = new SportsCar();
        sportsCar.start();
        sportsCar.driveFast();
        Bike bike = new Bike();
        bike.start();
    }
}

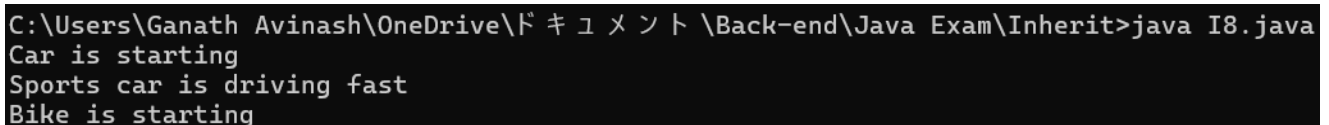
class Vehicle {
    void start() {
        System.out.println("Vehicle is starting");
    }
}

class Car extends Vehicle {
    void start() {
        System.out.println("Car is starting");
    }
}

class SportsCar extends Car {
    void driveFast() {
        System.out.println("Sports car is driving fast");
    }
}

class Bike extends Vehicle {
    void start() {
        System.out.println("Bike is starting");
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Inherit>java I8.java
Car is starting
Sports car is driving fast
Bike is starting
```

EXP: 8,9,10,11	POLYMORPHISM	OOPS
----------------	--------------	------

8a)

Code:

```

public class C1{
public static void main (String[]args){
emp e1=new emp(123,"Gojo Satoru",27);
e1.di();
e1.emdi();
}}
class Person{
int age;
String name;
Person(String name,int age){
this.name=name;
this.age=age;
}
void di(){
System.out.println("Name is: "+name);
System.out.println("Age is: "+age);
}}
class emp extends Person{
int emid;
int yrss;
emp(int emid,String name,int age){
super(name,age);
this.emid=emid;
}
void emdi(){
System.out.println("Employee id is: "+emid);
}
}

```

Screenshot:

```

C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>javac C1.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>java C1.java
Name is: Gojo Satoru
Age is: 27
Employee id is: 123

```

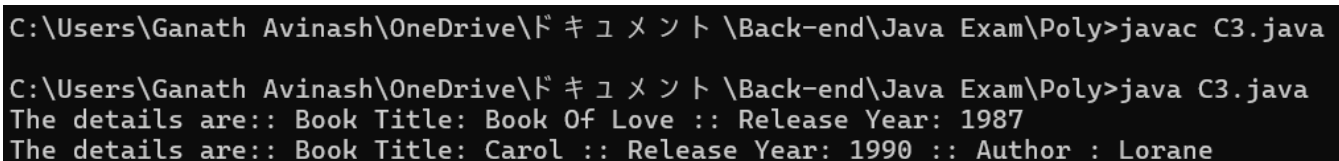
9a)

Code:

```
public class C3{
    public static void main(String[] args){
        Book b1=new Book(1987,"Book Of Love");
        Book b2=new Book(1990,"Carol","Lorane");
        b1.dis();
        b2.dis1();
    }
}
```

```
class Book{
    int yr;
    String title;
    String Author;
    Book(int yr,String title){
        this.title=title;
        this.yr=yr;
    }
    Book(int yr,String title,String Author){
        this.title=title;
        this.yr=yr;
        this.Author=Author;
    }
    void dis(){
        System.out.println("The details are:: Book Title: "+title+" :: Release Year: "+yr);
    }
    void dis1(){
        System.out.println("The details are:: Book Title: "+title+" :: Release Year: "+yr+" :: Author : "+Author);
    }
}
```

Screenshot:



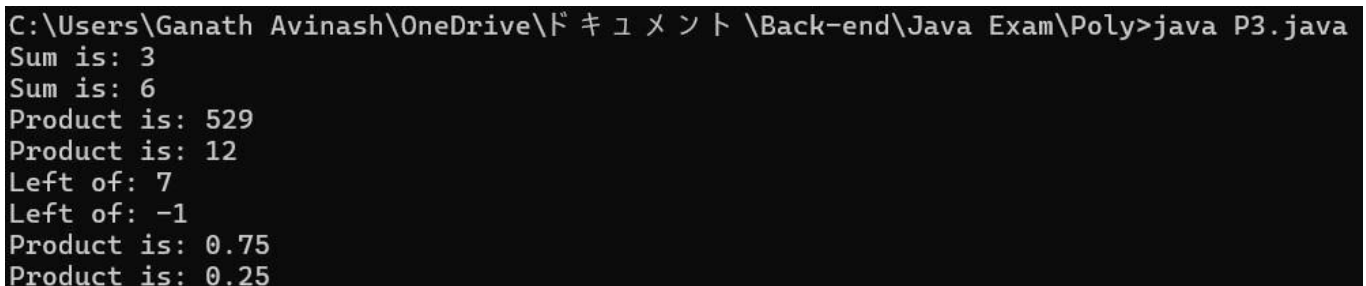
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>javac C3.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>java C3.java
The details are:: Book Title: Book Of Love :: Release Year: 1987
The details are:: Book Title: Carol :: Release Year: 1990 :: Author : Lorane
```


10a)

Code:

```
public class P3{
public static void main(String[]args){
Calculator c1=new Calculator();
c1.add(1,2);c1.add(1,2,3);
c1.mul(23,23);
c1.mul(2,3,2);
c1.sub(9,2);
c1.sub(8,4,5);
c1.div(3,4);
c1.div(3,4,3);
}}
class Calculator{
void add(int a,int b){
System.out.println("Sum is: "+(a+b));
}
void add(int a,int b,int c){
System.out.println("Sum is: "+(a+b+c));}
void mul(int a,int b){
System.out.println("Product is: "+(a*b));}
void mul(int a,int b,int c){
System.out.println("Product is: "+(a*b*c));}
void sub(int a,int b){
System.out.println("Left of: "+(a-b));}
void sub(int a,int b,int c){
System.out.println("Left of: "+(a-b-c));}
void div(double a,double b){
System.out.println("Product is: "+(a/b));}
void div(double a,double b,double c){
System.out.println("Product is: "+((a/b)/c));
}}
```

Screenshot:



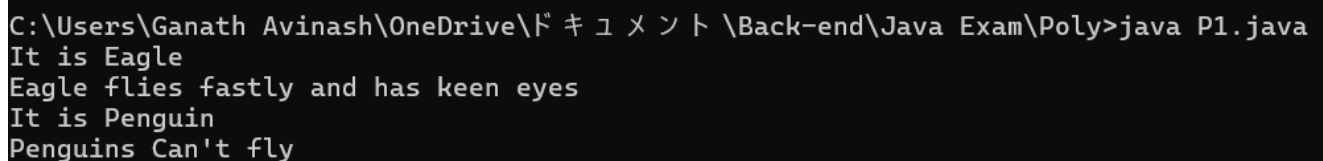
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>java P3.java
Sum is: 3
Sum is: 6
Product is: 529
Product is: 12
Left of: 7
Left of: -1
Product is: 0.75
Product is: 0.25
```

11a)

Code:

```
public class P1{
public static void main(String[]args){
eagle e1=new eagle("Eagle");
pen p1=new pen("Penguin");
e1.dis();e1.fly();
p1.dis();p1.fly();
}}
class Bird{
String type;
void fly(){
System.out.println("Birds fly in various speeds");
}
void dis(){
System.out.println("It is "+type);
}
}
class eagle extends Bird{
eagle(String typ){
super.type=typ;}
void fly(){
System.out.println("Eagle flies fastly and has keen eyes");
}
}
class pen extends Bird{
pen(String typ){
super.type=typ;
}
void fly(){
System.out.println("Penguins Can't fly");
}
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>java P1.java
It is Eagle
Eagle flies fastly and has keen eyes
It is Penguin
Penguins Can't fly
```

11b)

Code:

```
public class I8 {
    public static void main(String[] args) {
        SportsCar sportsCar = new SportsCar();
        sportsCar.start();
        sportsCar.driveFast();
        Bike bike = new Bike();
        bike.start(); }

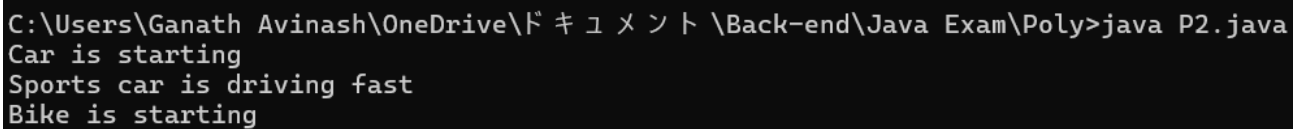
class Vehicle {
    void start() {
        System.out.println("Vehicle is starting");
    }
}

class Car extends Vehicle {
    void start() {
        System.out.println("Car is starting");
    }
}

class SportsCar extends Car {
    void driveFast() {
        System.out.println("Sports car is driving fast");
    }
}

class Bike extends Vehicle {
    void start() {
        System.out.println("Bike is starting");
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Poly>java P2.java
Car is starting
Sports car is driving fast
Bike is starting
```

EXP: 12,13	ABSTRACTION	OOPS
-------------------	--------------------	-------------

12a)

Code:

```
public class A1{
public static void main(String[]args){
Add a1=new Add();
Sub s1=new Sub();
Mul m1=new Mul();
Div d1=new Div();
a1.add(2,3);
s1.sub(3,4);
m1.mul(8,7);
d1.div(6,6);
}
}
```

```
abstract class Calculator{
abstract void add(int a,int b);
abstract void mul(int a,int b);
abstract void sub(int a,int b);
abstract void div(int a,int b);
}
```

```
class Add extends Calculator{
void add(int a,int b){
System.out.println("Sum is: "+(a+b));
}
void sub(int a,int b){}
void mul(int a,int b){}
void div(int a,int b){}
}
```

```
class Sub extends Calculator{
void add(int a,int b){}
void sub(int a,int b){
System.out.println("Subtracted value is: "+(a+b));
}
void mul(int a,int b){}
void div(int a,int b){}
}
```

```
class Mul extends Calculator{
void add(int a,int b){}
void sub(int a,int b){}
```

```
void mul(int a,int b){
System.out.println("Product is: "+(a+b));
}
void div(int a,int b){}
}
```

```
class Div extends Calculator{
void add(int a,int b){}
void sub(int a,int b){}
void mul(int a,int b){}
void div(int a,int b){
System.out.println("Divided answer is: "+(a+b));
}
}
```

Screenshot:

A screenshot of a command prompt window showing the execution of a Java program. The command 'java A1.java' has been run, and the output displays four lines of text: 'Sum is: 5', 'Subtracted value is: 7', 'Product is: 15', and 'Divided answer is: 12'.

```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A1.java
Sum is: 5
Subtracted value is: 7
Product is: 15
Divided answer is: 12
```

12b)

Code:

```
public class A2{
public static void main(String[]args){
Circle c1=new Circle(3);c1.calc();
Rect r1=new Rect(3,4);r1.calc();
Square s1=new Square(2);
s1.calc();}}
abstract class Shape{
abstract void calc();
void dis(){
System.out.println("Its a shape");
}}
class Circle extends Shape{
int a;
Circle(int a){
this.a=a;
}void calc(){
System.out.println("Area is: "+(3.14*a));
}}
class Square extends Shape{
int a;
Square(int a){
this.a=a;
}
void calc(){
System.out.println("Area is: "+(a*a));}}
class Rect extends Shape{
int a;int b;
Rect(int a,int b){
this.a=a;this.b=b;}
void calc(){
System.out.println("Area is: "+(b*a));
}}
```

Screenshot:



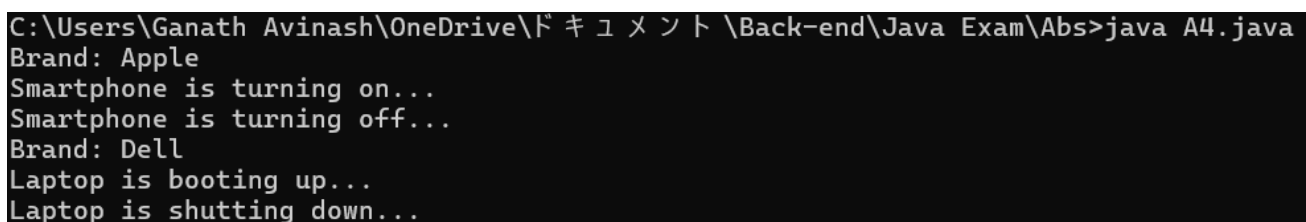
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A2.java
Area is: 9.42
Area is: 12
Area is: 4
```

12C)

Code:

```
public class A4{
    public static void main(String[] args) {
        ElectronicDevice phone = new Smartphone();
        ElectronicDevice laptop = new Laptop();
        phone.displayBrand("Apple");
        phone.turnOn();
        phone.turnOff();
        laptop.displayBrand("Dell");
        laptop.turnOn();
        laptop.turnOff(); } }
abstract class ElectronicDevice {
    abstract void turnOn();
    abstract void turnOff();
    void displayBrand(String brand) {
        System.out.println("Brand: " + brand);
    }
}
class Smartphone extends ElectronicDevice {
    void turnOn() {
        System.out.println("Smartphone is turning on...");
    }
    void turnOff() {
        System.out.println("Smartphone is turning off...");
    }
}
class Laptop extends ElectronicDevice {
    void turnOn() {
        System.out.println("Laptop is booting up...");
    }
    void turnOff() {
        System.out.println("Laptop is shutting down...");
    }
}
```

Screenshot:



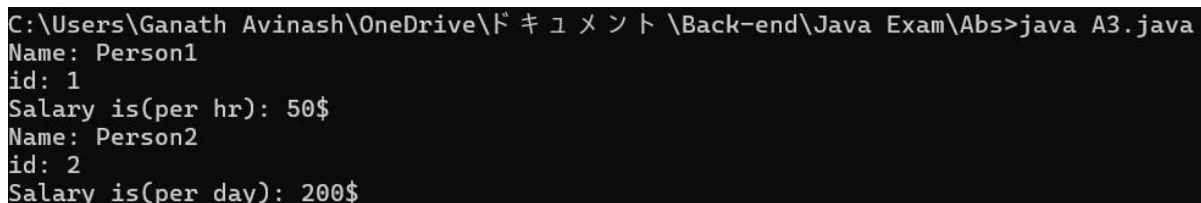
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A4.java
Brand: Apple
Smartphone is turning on...
Smartphone is turning off...
Brand: Dell
Laptop is booting up...
Laptop is shutting down...
```

12d)

Code:

```
public class A3{
public static void main(String[]args){
part p1=new part("Person1",1);
full f1=new full("Person2",2);
p1.dis();p1.cs();
f1.dis();f1.cs();
}}
abstract class Emp{
abstract void cs();
int id;
String Name;
Emp(String Name,int id){
this.Name=Name;
this.id=id;
}
void dis(){
System.out.println("Name: "+Name);
System.out.println("id: "+id);
}}
class part extends Emp{
part(String nam,int id){
super(nam,id);
}
void cs(){
System.out.println("Salary is(per hr): 50$");
}}
class full extends Emp{
full(String nam,int id){
super(nam,id);}
void cs(){
System.out.println("Salary is(per day): 200$");
}}
```

Screenshot:



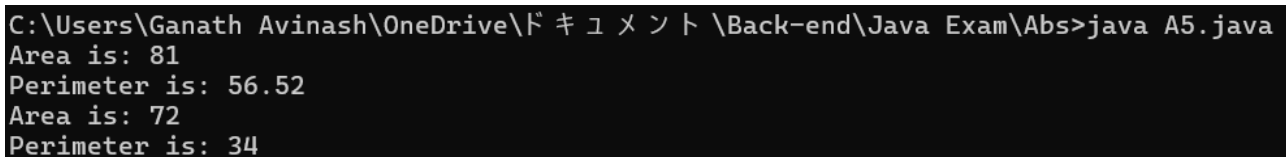
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A3.java
Name: Person1
id: 1
Salary is(per hr): 50$
Name: Person2
id: 2
Salary is(per day): 200$
```


13a)

Code:

```
public class A5{
public static void main(String[]args){
circle c1=new circle(9);
rect r1=new rect(8,9);
c1.area();c1.per();
r1.area();r1.per();
}}
interface shape{
void area();
void per();
}
class circle implements shape{
int a;
circle(int a){
this.a=a;
}
public void area(){
System.out.println("Area is: "+(a*a));
}
public void per(){
System.out.println("Perimeter is: "+(2*3.14*a));
}}
class rect implements shape{
int a;int b;
rect(int a,int b){
this.a=a;
this.b=b;
}
public void area(){
System.out.println("Area is: "+(a*b));
}
public void per(){
System.out.println("Perimeter is: "+(2*(a+b)));
}
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A5.java
Area is: 81
Perimeter is: 56.52
Area is: 72
Perimeter is: 34
```

13b)

Code:

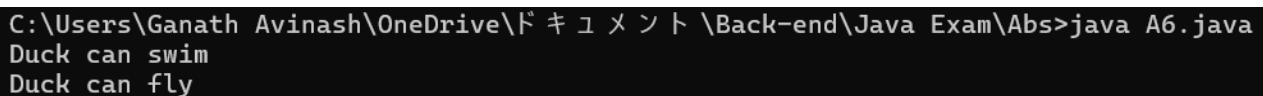
```
public class A6{
public static void main(String[]args){
duck d1=new duck();
d1.swim();
d1.fly();
}
}

interface swims{
void swim();
}

interface flyable{
void fly();
}

class duck implements swims,flyable{
public void swim(){
System.out.println("Duck can swim");
}
public void fly(){
System.out.println("Duck can fly");
}
}
```

Screenshot:

A screenshot of a Windows command prompt window. The title bar is not visible. The text in the window is as follows:
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A6.java
Duck can swim
Duck can fly
The text is displayed in a monospaced font, typical of a terminal or command prompt.

```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A6.java
Duck can swim
Duck can fly
```

13c)

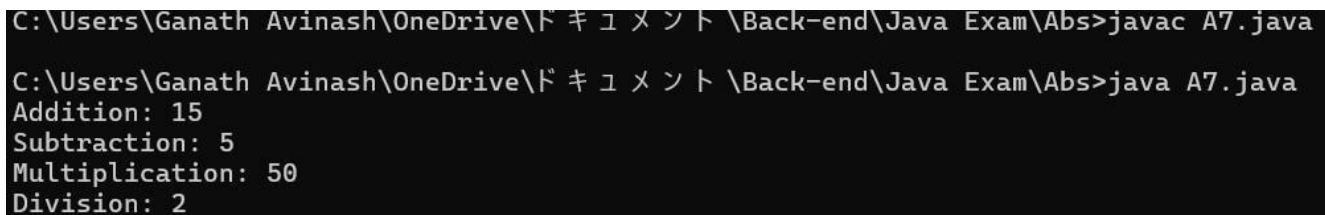
Code:

```
public class A7 {
    public static void main(String[] args) {
        Calculator add = (a, b) -> a + b;
        Calculator subtract = (a, b) -> a - b;
        Calculator multiply = (a, b) -> a * b;
        Calculator divide = (a, b) -> a / b;

        System.out.println("Addition: " + add.calculate(10, 5));
        System.out.println("Subtraction: " + subtract.calculate(10, 5));
        System.out.println("Multiplication: " + multiply.calculate(10, 5));
        System.out.println("Division: " + divide.calculate(10, 5));
    }
}

interface Calculator {
    int calculate(int a, int b);
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>javac A7.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A7.java
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2
```

13d)

Code:

```
public class A8{
public static void main(String[]args) {
Dog dog = new Dog();
dog.eat();
dog.sleep();
dog.makeSound();
}
}
```

```
interface Animal {
void eat();
void sleep();
default void makeSound() {
System.out.println("Animal sound");
}
}
```

```
class Dog implements Animal {
public void eat() {
System.out.println("Dog is eating.");
}
public void sleep() {
System.out.println("Dog is sleeping.");
}
public void makeSound() {
System.out.println("Bark");
}
}
```

Screenshot:

```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>javac A7.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>javac A8.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Abs>java A8.java
Dog is eating.
Dog is sleeping.
Bark
```

EXP: 14	ENCAPSULATION	OOPS
----------------	----------------------	-------------

14a)

Code:

```
public class E1{
public static void main(String[]args){
Person p1=new Person();
p1.set("Ganath",18);
p1.disp();
}
}
```

```
class Person{
private String name;
private int age;
void set(String a,int b){
name=a;
age=b;
}
void disp(){
System.out.println("Person name is: "+name+" and age is: "+age);
}
}
```

Screenshot:

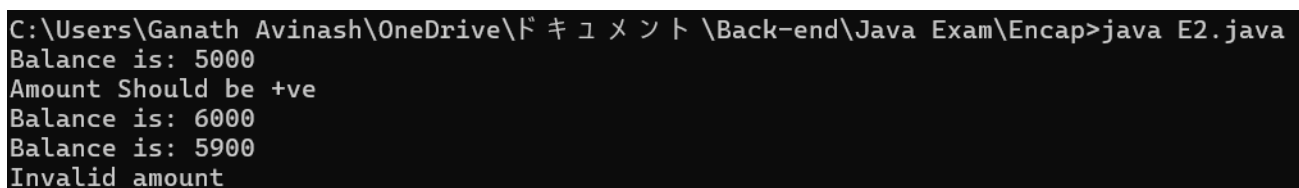
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>javac E1.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E1.java
Person name is: Ganath and age is: 18
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>|
```

14b)

Code:

```
public class E2{
public static void main(String[]args){
Bankacc b1=new Bankacc(123,5000);
b1.show();
b1.deposit(-1);
b1.deposit(1000);
b1.withdraw(100);
b1.withdraw(8000);}}
class Bankacc{
private int accno;
private long balance;
Bankacc(int accno,long balance){
this.accno=accno;
this.balance=balance;
}
void deposit(int a){
if (a>0){
balance+=a;
System.out.println("Balance is: "+balance);
}
else{
System.out.println("Amount Should be +ve");
}}
void withdraw(int a){
if (a>0 && balance>=a){
balance-=a;
System.out.println("Balance is: "+balance);
}
else{
System.out.println("Invalid amount");
}}
void show(){
System.out.println("Balance is: "+balance);
}}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E2.java
Balance is: 5000
Amount Should be +ve
Balance is: 6000
Balance is: 5900
Invalid amount
```

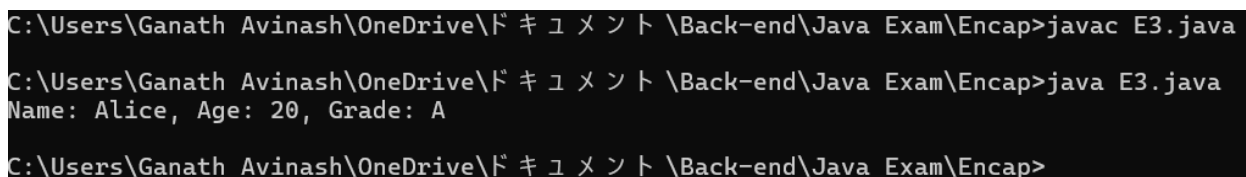
14c)

Code:

```
public class E3 {
    public static void main(String[] args) {
        Student student = new Student();
        student.setName("Alice");
        student.setAge(20);
        student.setGrade('A');
        System.out.println("Name: " + student.getName() + ", Age: " + student.getAge() + ", Grade: " +
            student.getGrade());
    }
}

class Student {
    private String name;
    private int age;
    private char grade;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        if (age > 0) {
            this.age = age;
        } else {
            System.out.println("Age cannot be negative.");
        }
    }
    public char getGrade() {
        return grade;
    }
    public void setGrade(char grade) {
        this.grade = grade;
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>javac E3.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E3.java
Name: Alice, Age: 20, Grade: A
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>
```

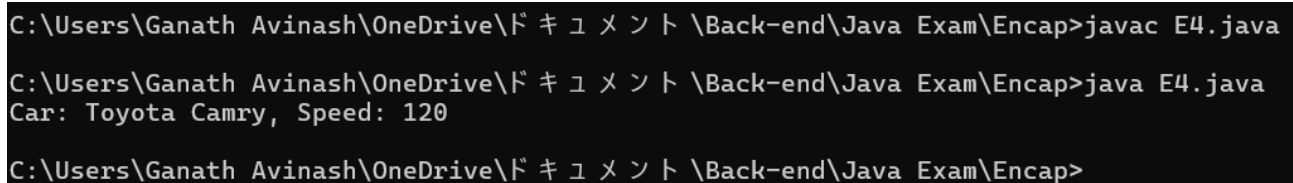
14d)

Code:

```
public class E4 {
    public static void main(String[] args) {
        Car car = new Car("Toyota", "Camry");
        car.setSpeed(120);
        System.out.println("Car: " + car.getBrand() + " " + car.getModel() + ", Speed: " +
car.getSpeed());    }}

class Car {
    private String brand;
    private String model;
    private int speed;
    public Car(String brand, String model) {
        this.brand = brand;
        this.model = model;
        this.speed = 0; // Default speed
    }
    public String getBrand() {
        return brand;
    }
    public String getModel() {
        return model;
    }
    public int getSpeed() {
        return speed;
    }
    public void setSpeed(int speed) {
        if (speed >= 0 && speed <= 200) {
            this.speed = speed;
        } else {
            System.out.println("Speed must be between 0 and 200.");
        }
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>javac E4.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E4.java
Car: Toyota Camry, Speed: 120
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>
```

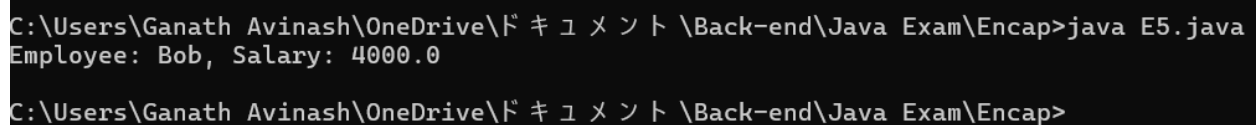

14e)

Code:

```
public class E5 {
    public static void main(String[] args) {
        Employee emp = new Employee(101, "Bob", 4000);
        System.out.println("Employee: " + emp.getName() + ", Salary: " + emp.getSalary());
    }
}

class Employee {
    private int id;
    private String name;
    private double salary;
    public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        setSalary(salary);
    }
    public int getId() {
        return id;
    }
    public String getName() {
        return name;
    }
    public double getSalary() {
        return salary;
    }
    public void setSalary(double salary) {
        if (salary >= 3000) {
            this.salary = salary;
        } else {
            System.out.println("Salary must be at least 3000.");
        }
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E5.java
Employee: Bob, Salary: 4000.0
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>
```

14f)

Code:

```
public class E6 {
    public static void main(String[] args) {
        Patient patient = new Patient(101, "David", "Flu");
        System.out.println("Patient ID: " + patient.getPatientID() + ", Name: " + patient.getName()
+ ", Disease: " + patient.getDisease());
    }
}
class Patient {
    private final int patientID;
    private String name;
    private String disease;

    public Patient(int patientID, String name, String disease) {
        this.patientID = patientID;
        this.name = name;
        this.disease = disease;
    }

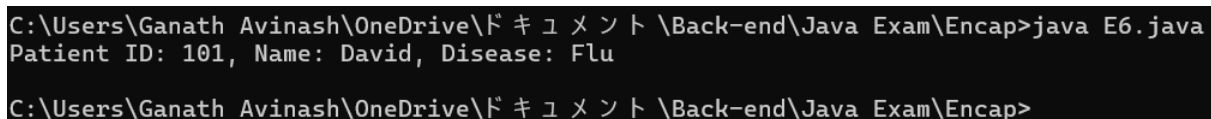
    public int getPatientID() {
        return patientID;
    }

    public String getName() {
        return name;
    }

    public String getDisease() {
        return disease;
    }

    public void setDisease(String disease) {
        this.disease = disease;
    }
}
```

Screenshot:



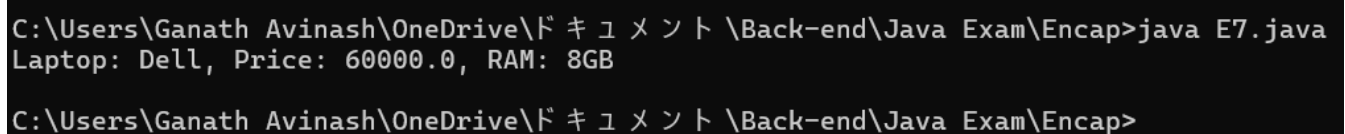
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E6.java
Patient ID: 101, Name: David, Disease: Flu
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>
```

14g)

Code:

```
public class E7{
    public static void main(String[] args) {
        Laptop laptop = new Laptop("Dell", 60000, 8);
        System.out.println("Laptop: " + laptop.getBrand() + ", Price: " + laptop.getPrice() + ", RAM: " + laptop.getRAM() + "GB");
    }
}
class Laptop {
    private String brand;
    private double price;
    private int RAM;
    public Laptop(String brand, double price, int RAM) {
        this.brand = brand;
        setPrice(price);
        setRAM(RAM);
    }
    public String getBrand() {
        return brand;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        if (price > 0) {
            this.price = price;
        } else {
            System.out.println("Price must be positive.");
        }
    }
    public int getRAM() {
        return RAM;
    }
    public void setRAM(int RAM) {
        if (RAM >= 4) {
            this.RAM = RAM;
        } else {
            System.out.println("RAM must be at least 4GB.");
        }
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E7.java
Laptop: Dell, Price: 60000.0, RAM: 8GB
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>
```

14h)

Code:

```
public class E8 {
    public static void main(String[] args) {
        Person person = new Person();
        person.setFirstName("Emily");
        person.setLastName("Clark");
        person.setAge(30);
        person.dis();
        System.out.println("Person Age: " + person.getAge());
    }
}
class Person {
    private String firstName;
    private String lastName;
    private int age;
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public void dis() {
        System.out.println("Person 1st name: " +firstName );
        System.out.println("Person Last name: " +lastName );
    }
    public void setAge(int age) {
        if (age > 0 && age <= 120) {
            this.age = age;
        } else {
            System.out.println("Age must be between 1 and 120.");
        }
    }
    public int getAge() {
        return age;
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java E8.java
Person 1st name: Emily
Person Last name: Clark
Person Age: 30
```

15a)

Code:

```
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.util.Random;
import java.util.Date;
import java.io.FileWriter;
import java.io.IOException;

public class Builtin {
    public static void main(String[] args) {
        LocalDate today = LocalDate.now();
        LocalDateTime now = LocalDateTime.now();
        System.out.println("Today's Date: " + today);
        System.out.println("Current Time: " + now);

        Random random = new Random();
        int randNumber = random.nextInt(100);
        System.out.println("Random Number: " + randNumber);

        // File
        try {
            FileWriter writer = new FileWriter("output.txt");
            writer.write("Date: " + today + "\n");
            writer.write("Time: " + now + "\n");
            writer.write("Random Number: " + randNumber + "\n");
            writer.close();
            System.out.println("Data successfully written to file.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Screenshot:

Output.txt

```
date: 2025-03-25
Time: 21:43:23.477120300
Random Number: 7
```

```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java Builtin.java
Today's Date: 2025-03-25
Current Time: 21:43:23.477120300
Random Number: 7
Data successfully written to file.
```

15b)

Code:

```
import java.awt.*;
import java.awt.event.*;
import java.util.Date;

public class BuiltIn2 {
    public static void main(String[] args) {

        JFrame frame = new JFrame("Date & Time Display");
        frame.setSize(400, 200);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        JLabel label = new JLabel("Click the button to see date & time!");
        label.setBounds(50, 30, 300, 30);
        frame.add(label);

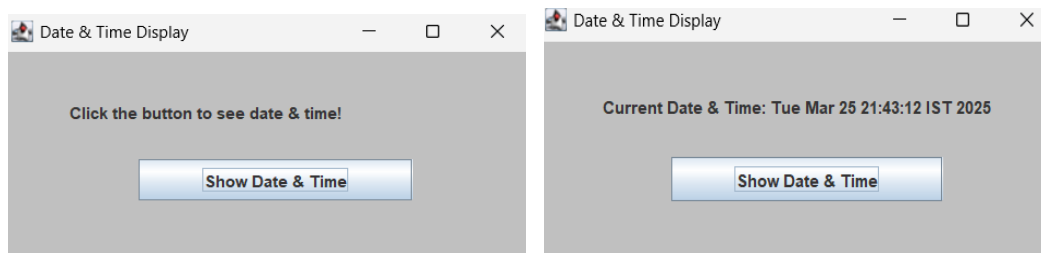
        JButton button = new JButton("Show Date & Time");
        button.setBounds(100, 80, 200, 30);
        frame.add(button);

        frame.getContentPane().setBackground(Color.LIGHT_GRAY);

        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Date now = new Date();
                label.setText("Current Date & Time: " + now.toString());
            }
        });

        frame.setVisible(true);
    }
}
```

Screenshot:



15c)

Code:

Program 1:

```
package calculator;
public class Calculator {
    public int add(int a, int b) {
        return a + b;    }
    public int subtract(int a, int b) {
        return a - b;    }
    public int multiply(int a, int b) {
        return a * b;    }
    public double divide(int a, int b) {
        if (b == 0) {
            System.out.println("Error: Cannot divide by zero.");
            return 0;
        }
        return (double) a / b;    }}

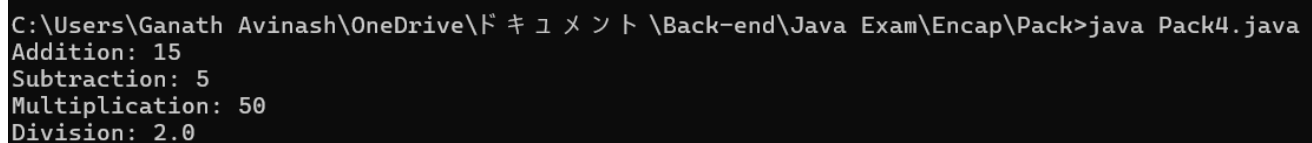
```

Program 2:

```
import calculator.Calculator;
public class Pack4{
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Addition: " + calc.add(10, 5));
        System.out.println("Subtraction: " + calc.subtract(10, 5));
        System.out.println("Multiplication: " + calc.multiply(10, 5));
        System.out.println("Division: " + calc.divide(10, 5));    }}

```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap\Pack>java Pack4.java
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0

```

15d) Code:

Program 1:

```
package shapes;
public class Circle {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
}
```

Program 2:

```
package shapes;
public class Rectangle {
    private double length, width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public double getArea() {
        return length * width;
    }
}
```

Program 3:

```
import shapes.Circle;
import shapes.Rectangle;
public class Pack2 {
    public static void main(String[] args) {
        Circle c = new Circle(5);
        Rectangle r = new Rectangle(4, 6);
        System.out.println("Circle Area: " + c.getArea());
        System.out.println("Rectangle Area: " + r.getArea());
    }
}
```

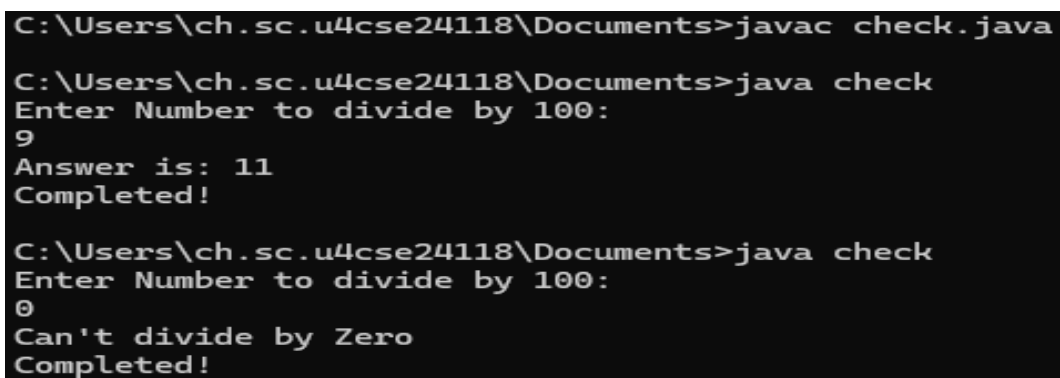
Screenshot:

```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap\Pack>java Pack2.java
Circle Area: 78.53981633974483
Rectangle Area: 24.0
```


16a)

Code:

```
import java.util.Scanner;
class Divide{
void div(int a){
try{
int ff=100/a;
System.out.println("Answer is: "+ff);
}
catch(Exception e){
System.out.println("Can't divide by Zero");
}}}
public class check{
public static void main(String[]args){
Divide d1=new Divide();
Scanner ip = new Scanner(System.in);
System.out.println("Enter Number to divide by 100: ");
try{
int num=ip.nextInt();
d1.div(num);
}
catch(Exception e){
System.out.println("Enter only number!!");
}
finally{
System.out.println("Completed!");
}}}
```

Screenshot:

```
C:\Users\ch.sc.u4cse24118\Documents>javac check.java
C:\Users\ch.sc.u4cse24118\Documents>java check
Enter Number to divide by 100:
9
Answer is: 11
Completed!

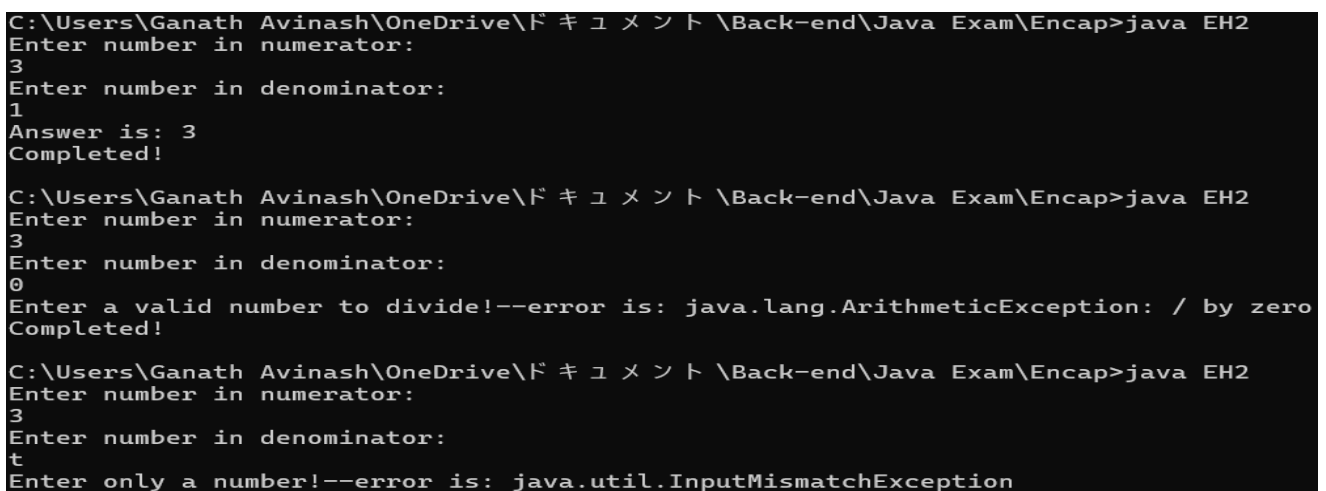
C:\Users\ch.sc.u4cse24118\Documents>java check
Enter Number to divide by 100:
0
Can't divide by Zero
Completed!
```

16b)

Code:

```
import java.util.Scanner;
import java.util.InputMismatchException;
class Two{
void divide(int a,int b){
try{
int ans=a/b;
System.out.println("Answer is: "+ans);
}
catch(ArithmeticException error){
System.out.println("Enter a valid number to divide!--error is: "+error);
}
finally{System.out.println("Completed!");}}
public class EH2{
public static void main(String[]args){
Two t1=new Two();
Scanner ip=new Scanner(System.in);
try{
System.out.println("Enter number in numerator: ");
int a=ip.nextInt();
System.out.println("Enter number in denominator: ");
int b=ip.nextInt();
t1.divide(a,b);
}
catch(InputMismatchException e){
System.out.println("Enter only a number!--error is: "+e);
}}}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java EH2
Enter number in numerator:
3
Enter number in denominator:
1
Answer is: 3
Completed!

C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java EH2
Enter number in numerator:
3
Enter number in denominator:
0
Enter a valid number to divide!--error is: java.lang.ArithmeticException: / by zero
Completed!

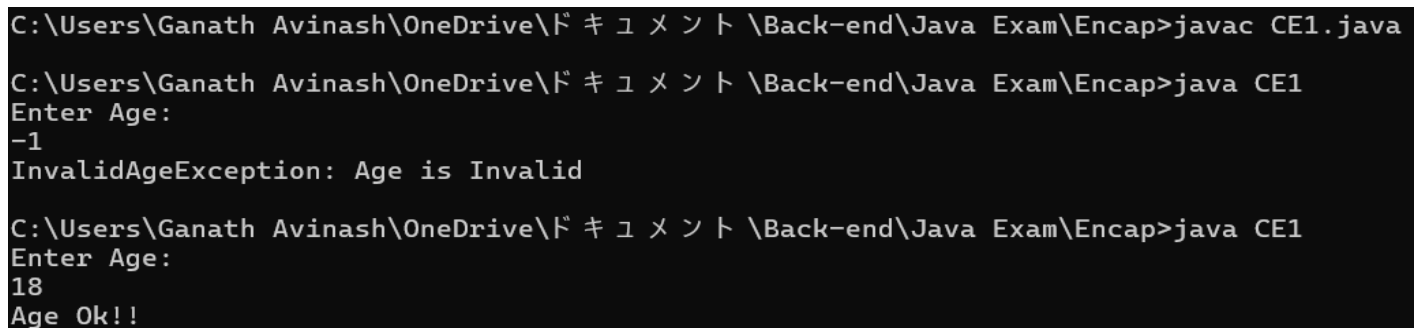
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java EH2
Enter number in numerator:
3
Enter number in denominator:
t
Enter only a number!--error is: java.util.InputMismatchException
```

16c)

Code:

```
import java.util.Scanner;
class InvalidAgeException extends Exception {
    InvalidAgeException(String message) {
        super(message);
    }
}
class Agev {
    void checkAge(int a) {
        try {
            if (a < 0 || a > 150) {
                throw new InvalidAgeException("Age is Invalid");
            }
        }
        catch (InvalidAgeException e) {
            System.out.println(e);
        }
        System.out.println("Age Ok!!");
    }
}
public class CE1 {
    public static void main(String[] args) {
        Agev a1 = new Agev();
        Scanner ip = new Scanner(System.in);
        System.out.println("Enter Age: ");
        int a = ip.nextInt();
        a1.checkAge(a);
        ip.close();
    }
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>javac CE1.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java CE1
Enter Age:
-1
InvalidAgeException: Age is Invalid
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java CE1
Enter Age:
18
Age Ok!!
```

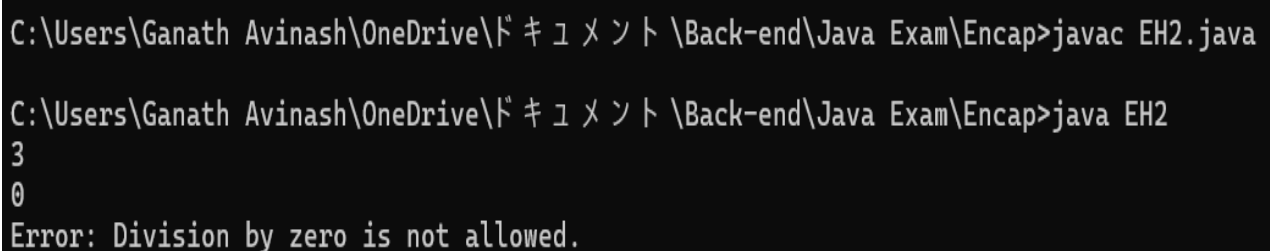
16d)

Code:

```
import java.util.Scanner;
public class EH2 {
    public void divide(int a, int b) throws ArithmeticException {
        int result= a / b;
    }

    public static void main(String[] args) {
        try {
            Scanner ip=new Scanner(System.in);
            int a=ip.nextInt();
            int b=ip.nextInt();
            EH2 e1= new EH2();
            e1.divide(a,b);
            System.out.println("Result: " + a/b);
        }
        catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }
    }
}
```

Screenshot:



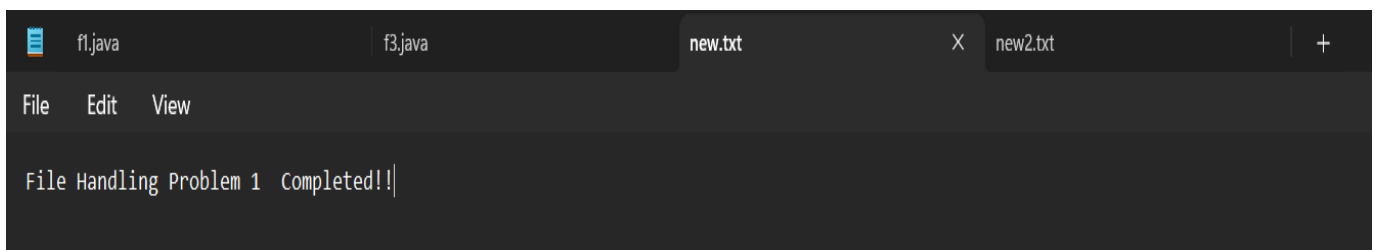
```
C:\Users\Ganath Avinash\OneDrive\ト キ ュ メ ント \Back-end\Java Exam\Encap>javac EH2.java
C:\Users\Ganath Avinash\OneDrive\ト キ ュ メ ント \Back-end\Java Exam\Encap>java EH2
3
0
Error: Division by zero is not allowed.
```

17a) Writing Using FileWriter

Code:

```
import java.io.FileWriter;
public class f1{
    public static void main(String[]args){
        try{
            FileWriter f1=new FileWriter("new.txt");
            f1.write("File Handling Problem 1");
            f1.append(" Completed!!");
            f1.close();
        }
        catch(Exception error){
            System.out.println("Error is: "+error);
        }
    }
}
```

Screenshot:

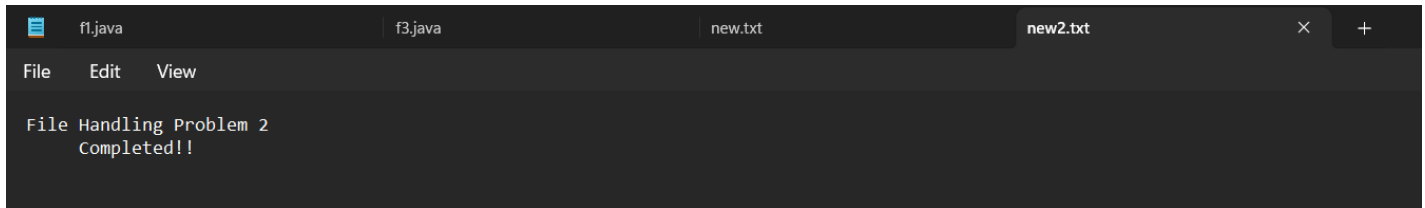


17b) Writing Using Buffer

Code:

```
import java.io.FileWriter;
import java.io.BufferedWriter;
import java.util.Scanner;
public class f3{
    public static void main(String[]args){
        try{
            FileWriter f1=new FileWriter("new2.txt");
            BufferedWriter b1= new BufferedWriter(f1);
            b1.write("File Handling Problem 2");
            b1.newLine();
            b1.write(" Completed!!");
            b1.close();}
        catch(Exception error){
            System.out.println("Error is: "+error);
        }
    }
}
```

Screenshot:



17c) Reading Using FileReader

Code:

```
import java.io.FileReader;
import java.io.FileNotFoundException;
public class f4{
    public static void main(String[]args){
        try{
            FileReader f1=new FileReader("Mera.txt");
            int c=f1.read();
            while (c!=-1){
                System.out.print((char)c);
                c=f1.read();
            }
            f1.close();
        }
        catch(Exception e){
            System.out.println("Error found");
        }
    }
}
```

Screenshot:

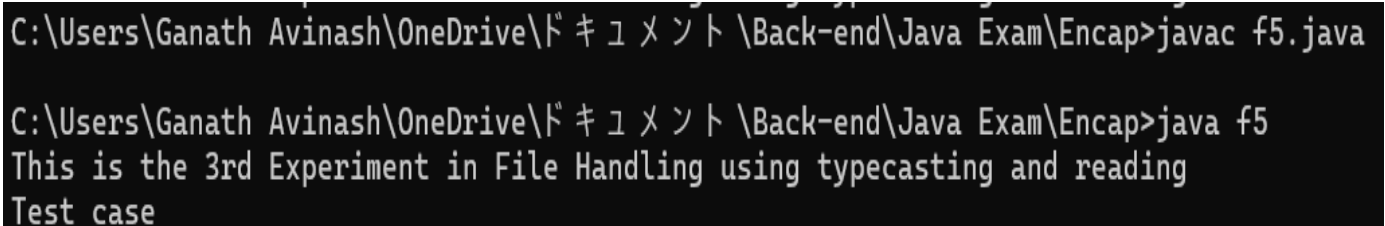
```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>javac f4.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java f4
This is the 3rd Experiment in File Handling using typecasting and reading
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>
```

17d) Same Experiment With Buffered Reader (fast and efficient)

Code:

```
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
public class f5{
public static void main(String[]args){
try{
FileReader f1=new FileReader("Mera.txt");
BufferedReader b1=new BufferedReader(f1);
String c=b1.readLine();
while (c!=null){
System.out.println(c);
c=b1.readLine();
}
b1.close();
}
catch(Exception e){
System.out.println("Error found");
}
}
}
```

Screenshot:



```
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>javac f5.java
C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\Java Exam\Encap>java f5
This is the 3rd Experiment in File Handling using typecasting and reading
Test case
```

.....

THANKYOU

By Ganath Avinash G.R

CH.SC.U4CSE24118