1.

public class room {  
 int roomno;  
 String roomtype;  
 double roomarea;  
 boolean ACmachine;  
  
 public void setdata(int roomno, String roomtype, double roomarea, boolean ACmachine){  
 this.roomno = roomno;  
 this.roomtype = roomtype;  
 this.roomarea = roomarea;  
 this.ACmachine = ACmachine;  
 }  
  
 public void displaydata(){  
 System.out.println("Room no: " + this.roomno + "\nRoom type: " + this.roomtype + "\nRoom area: " + this.roomarea + "\nAC: " + this.ACmachine);  
 }  
  
 public static void main(String[] args) {  
 room obj = new room();  
  
 obj.setdata(611, "Bad Room", 800, true);  
 obj.displaydata();  
 }  
}

2.

public class simpleobject {  
  
 public simpleobject(){  
 System.out.println("Hello, simple object is created.");  
 }  
  
 public static void main(String[] args) {  
 simpleobject obj = new simpleobject();  
  
 }  
}

3.

public class This {  
 private String name;  
 private String id;  
  
 public This(String name, String id){  
 this.name = name;  
 this.id = id;  
 }  
  
 public void display(){  
 System.out.println(this.name + "\n" + this.id);  
 }  
  
 public static void main(String[] args) {  
 This obj = new This("Shrikanta Paul", "221-15-4868");  
  
 obj.display();  
 }  
}

4.

public class Car {  
 public String model;  
  
 public Car(){  
 System.out.println("Hello.");  
 }  
  
 public void display(){  
 System.out.println(this.model);  
 }  
}  
  
class BMW extends Car{  
  
 public BMW(){  
 super();  
 super.model = "BMW";  
 super.display();  
 }  
 public static void main(String[] args) {  
 BMW obj = new BMW();  
 }  
}

5.

public class OverLoading {  
  
 public void area(int a){  
 System.out.println("Area of Square(in int): " + a\*a);  
 }  
  
 public void area(int a, int b){  
 System.out.println("Area of Rectangle: " + a\*b);  
 }  
  
 public void area(float a){  
 System.out.println("Area of Square(in float): " + a\*a);  
 }  
  
 public static void main(String[] args) {  
 OverLoading obj = new OverLoading();  
   
 obj.area(4);  
 obj.area(3.3f);  
 obj.area(5, 6);  
 }  
}

6.

public class a {  
  
 public void function(){  
 System.out.println("Can eat.");  
 }  
}  
  
  
class b extends a{  
  
 public void function(){  
 System.out.println("Can fly.");  
 }  
  
 public static void main(String[] args) {  
 b obj = new b();  
  
 obj.function();  
 }  
}

7.

public class shape {  
  
 public void draw() {  
 System.out.println("Draw the shape.");  
 }  
  
 public void erase() {  
 System.out.println("Erase the shape.\n");  
 }  
}  
  
  
class circle extends shape {  
 public void draw() {  
 System.out.println("Draw the Circle.");  
 }  
  
 public void erase() {  
 System.out.println("Erase the Circle.\n");  
 }  
}  
  
  
class triangle extends shape {  
 public void draw() {  
 System.out.println("Draw the Triangle.");  
 }  
  
 public void erase() {  
 System.out.println("Erase the Triangle.\n");  
 }  
}  
  
  
class square extends shape {  
 public void draw() {  
 System.out.println("Draw the Square.");  
 }  
  
 public void erase() {  
 System.out.println("Erase the Square.");  
 }  
}  
  
class Main{  
 public static void main(String[] args) {  
  
 circle c = new circle();  
 c.draw();  
 c.erase();  
  
 triangle t = new triangle();  
 t.draw();  
 t.erase();  
  
 square s = new square();  
 s.draw();  
 s.erase();  
 }  
}

8.

public abstract class Abstract {  
  
 public abstract void method1();  
  
 public void method2(){  
 System.out.println("Method 2.");  
 }  
}  
  
class NonAbstract extends Abstract{  
  
 public void method1(){  
 System.out.println("Method 1.");  
 }  
  
 public static void main(String[] args) {  
 NonAbstract obj = new NonAbstract();  
 obj.method1();  
 obj.method2();  
 }  
}

9.

public interface A {  
  
 void meth1();  
 void meth2();  
}  
  
class MyClass implements A{  
  
 public void meth1(){  
 System.out.println("Meth1.");  
 }  
  
 public void meth2(){  
 System.out.println("Meth2.");  
 }  
  
 public static void main(String[] args) {  
 MyClass obj = new MyClass();  
  
 obj.meth1();  
 obj.meth2();  
 }  
}

10.

import java.awt.\*;  
  
public interface Parent1 {  
 void methOfP1();  
}  
  
interface Parent2{  
 void methOfP2();  
}  
  
class Child implements Parent1, Parent2{  
  
 public void methOfP1(){  
 System.out.println("Method of parent1.");  
 }  
  
 public void methOfP2() {  
 System.out.println("Method of parent2.");  
 }  
  
 public static void main(String[] args) {  
 Child obj = new Child();  
 obj.methOfP1();  
 obj.methOfP2();  
 }  
}

11.

public interface test {  
 void square();  
}  
  
class arithmetic implements test{  
  
 public void square(){  
 System.out.println("This is square from arithmetic class.");  
 }  
}  
  
class ToTestInt extends arithmetic{  
 public static void main(String[] args) {  
 arithmetic obj = new arithmetic();  
 obj.square();  
 }  
}

12.

public class ColourChecking {  
  
 public static void main(String[] args) {  
  
 int red = 193;  
 int green = 255;  
 int blue = 183;  
  
 float r = red / 255f;  
 float g = green / 255f;  
 float b = blue / 255f;  
  
 float hue;  
 float saturation;  
 float brightness;  
  
 float cMax = Math.max(Math.max(r, g), b);  
 float cMin = Math.min(Math.min(r, g), b);  
 float delta = cMax - cMin;  
  
  
 // Calculate Hue  
 if (delta == 0) {  
 hue = 0;  
 }  
 else if (cMax == r) {  
 hue = ((g - b) / delta) % 6;  
 }  
 else if (cMax == g) {  
 hue = (b - r) / delta + 2;  
 }  
 else {  
 hue = (r - g) / delta + 4;  
 }  
  
 hue \*= 60;  
 if (hue < 0) {  
 hue += 360;  
 }  
  
 // Calculate Brightness  
 brightness = ((cMax + cMin) / 2) \* 100;  
  
 // Calculate Saturation  
 if (r == g || g == b || b == r) {  
 saturation = 0;  
 }  
 else if(brightness<128) {  
 saturation = 255 \* delta / (cMax + cMin);  
 }  
 else{  
 saturation = 255 \* delta / (511 - (cMax + cMin));  
 }  
  
  
  
 System.out.println("Hue: " + hue);  
 System.out.println("Saturation: " + saturation);  
 System.out.println("Brightness: " + brightness);  
 }  
}

13.

import java.util.\*;  
  
public class FibonacciRecursion {  
  
 public static void main(String[] args) {  
  
 Scanner in = new Scanner(System.in);  
 int n = in.nextInt();  
  
 System.out.println("Fibonacci Series:");  
  
 for (int i = 0; i < n; i++) {  
  
 System.out.print(fibonacci(i) + " ");  
 }  
 }  
  
 public static int fibonacci(int n) {  
 if (n <= 1) {  
 return n;  
 }  
 return fibonacci(n - 1) + fibonacci(n - 2);  
 }  
}

14.

package mypack;  
  
public class Point {  
 private int x,y;  
  
 public Point(){  
 System.out.println("x= " + this.x + "y= " + this.y);  
 }  
  
 Point(int x, int y){  
 this.x = x;  
 this.y = y;  
 }  
  
 public void setX(int x){  
  
 this.x = x;  
 }  
  
 public void setY(int y){  
  
 this.y = y;  
 }  
  
 public void setXY(int x, int y){  
  
 this.x = x;  
 this.y = y;  
 }  
}

import mypack.Point;  
public class Main {  
  
 public static void main(String[] args) {  
  
 Point obj = new Point();  
 Point obj2 = new Point(10, 20);  
  
 obj2.setX(100);  
 obj2.setY(200);  
 obj2.setXY(1000, 2000);  
 }  
}

15.

public class Box {  
  
 double length;  
 double breadth;  
 double height;  
 Box(){  
 System.out.println("This is Box.");  
 }  
  
 public void setLength(double length) {  
 this.length = length;  
 }  
  
 public void setBreadth(double breadth) {  
 this.breadth = breadth;  
 }  
  
 public void setHeight(double height) {  
 this.height = height;  
 }  
  
 public double area(){  
 return this.length \* this.height;  
 }  
  
 public double volume(){  
 return this.length \* this.breadth \* this.height;  
 }  
}  
  
class Box3d extends Box{  
 double length;  
 double breadth;  
 double height;  
 Box3d(double height, double breadth, double length){  
 super.setBreadth(breadth);  
 super.setHeight(height);  
 super.setLength(length);  
 System.out.println("This is Box Area: " + super.area());  
 System.out.println("This is Box Volume: " + super.volume());  
   
 System.out.println("\nThis is Box3d.");  
 }  
  
 public void setLength(double length) {  
 this.length = length;  
 }  
  
 public void setBreadth(double breadth) {  
 this.breadth = breadth;  
 }  
  
 public void setHeight(double height) {  
 this.height = height;  
 }  
  
 public double area(){  
 return this.length \* this.height;  
 }  
  
 public double volume(){  
 return this.length \* this.breadth \* this.height;  
 }  
}  
  
class Main{  
 public static void main(String[] args) {  
 Box3d obj = new Box3d(1, 2, 3);  
  
 obj.setLength(10);  
 obj.setBreadth(20);  
 obj.setHeight(30);  
 System.out.println("This is Box3d Area: " + obj.area());  
 System.out.println("This is Box3d Volume: " + obj.volume());  
 }  
}

16.

class Department {  
 private String name;  
  
 public Department(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
}  
  
  
class Professor {  
 private String name;  
  
 public Professor(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
}  
  
  
class Student {  
 private String name;  
  
 public Student(String name) {  
 this.name = name;  
 }  
  
 public String getName() {  
 return name;  
 }  
}  
  
  
class University {  
 private String name;  
 private Department department;  
 private Professor professor;  
 private Student student;  
  
  
 public void setProfessor(Professor professor) {  
 this.professor = professor;  
 }  
  
  
 public void setDepartment(Department department) {  
 this.department = department;  
 }  
  
  
 public void setStudent(Student student) {  
 this.student = student;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void displayUniversityInfo() {  
  
 System.out.println("University: " + name);  
 System.out.println("Department: " + department.getName());  
 System.out.println("Professor: " + professor.getName());  
 System.out.println("Student: " + student.getName());  
 }  
}  
  
  
class MainMethod {  
 public static void main(String[] args) {  
  
 Department department = new Department("CSE");  
 Professor professor = new Professor("XYZ");  
 Student student = new Student("Shrikanta Paul");  
  
  
 University university = new University();  
 university.setName("Daffodil International University");  
  
  
 university.setDepartment(department);  
 university.setProfessor(professor);  
 university.setStudent(student);  
  
  
 university.displayUniversityInfo();  
 }  
}

17.

public class NegativeArraySizeEx {  
  
 public static void main(String[] args) {  
  
 try{  
 int[] a = new int[-5];  
 }  
  
 catch(NegativeArraySizeException e){  
 System.out.println("This is Negative Array Size Exception from catch.\n");  
 }  
  
 finally {  
 System.out.println("The exception is Negative array size from finally.");  
 }  
 }  
}

18.

import java.util.\*;  
  
public class ThrowsEx {  
  
 public static void main(String[] args) {  
  
 Scanner in = new Scanner(System.in);  
  
 int number = in.nextInt();  
  
 if(number > 100 || number < 0){  
 throw new ArithmeticException("Not valid");  
 }  
 else if(number < 40){  
 System.out.println("Fail");  
 }  
 else{  
 System.out.println("Pass.");  
 }  
 }  
}

19.

import java.util.\*;  
  
public class UserDefinedEx extends Exception{  
  
 UserDefinedEx(String exception){  
  
 System.out.println(exception);  
  
 }  
  
 public static void main(String[] args) {  
  
 Scanner in = new Scanner(System.in);  
 try{  
 int a = in.nextInt();  
 int b = in.nextInt();  
 int c = in.nextInt();  
  
 if(a\*b\*c == 0){  
 throw new UserDefinedEx("The multiplication can't be zero.\n");  
 }  
  
 System.out.println("There is no Exception.");  
 }  
 catch(UserDefinedEx e){  
 System.out.println("This is a User Defined Exception.");  
 }  
 }  
}

20.

import java.util.\*;  
import java.util.Collections;  
  
public class ArrayListSorting {  
 public static void main(String[] args) {  
  
 ArrayList<String> numbers = new ArrayList<>();  
 numbers.add("Shrikanta");  
 numbers.add("Nurol");  
 numbers.add("Hira");  
 numbers.add("Shrabon");  
 numbers.add("Anik");  
  
 System.out.println("Unsorted ArrayList: " + numbers);  
  
 Collections.sort(numbers);  
  
 System.out.println("Sorted ArrayList: " + numbers);  
 }  
}