# LAB # 04

**Task No 01:** Write a program using the concepts of a default constructor. Consider a computer system whose name, type, processor specification, ram, hard disk drives, mother board, optical drive etc., in a default constructor, desired values are entered by the user in a get method (that takes information from the user) and the displays the inputted information via display method. The user shall be asked to change any of the provided information if he/she agrees to change the information then new values shall be asked from the user.

**Code:**

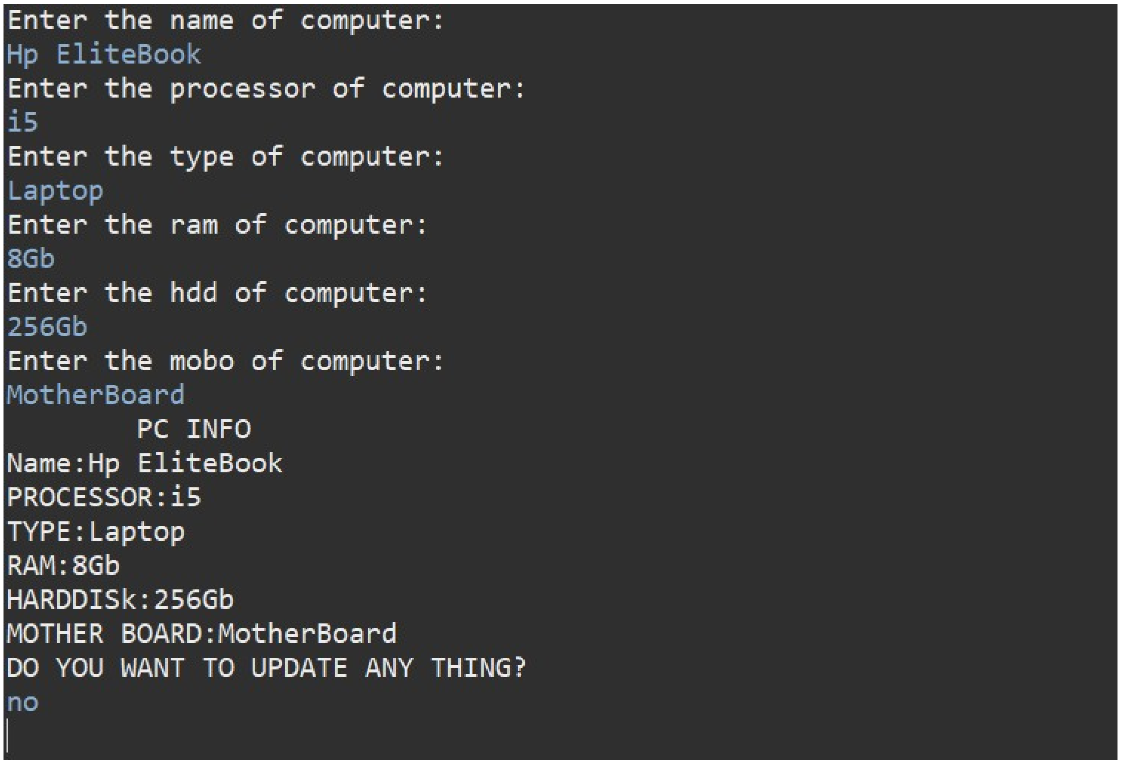
**Main:**

```
package lab04task01;

public class Lab04task01 {

    public static void main(String[] args) {

        Computer c = new Computer();

        c.setname();
        c.setcpu();
        c.settype();
        c.setram();
        c.sethdd();
        c.setmobo();
        c.display();
        c.update();
    }
}
```

**Computer:**

```
package lab04task01;

import java.util.Scanner;

public class Computer {

    private String name, processor, type, ram, hdd, mobo;
    Scanner sc = new Scanner(System.in);

    public void setname() {
        System.out.println("Enter the name of computer:");
        this.name = sc.nextLine();
    }
    public void setcpu() {
        System.out.println("Enter the processor of computer:");
        this.processor = sc.nextLine();
    }
    public void settype() {
        System.out.println("Enter the type of computer:");
        this.type = sc.nextLine();
    }
    public void setram() {
        System.out.println("Enter the ram of computer:");
        this.ram = sc.nextLine();
    }
    public void sethdd() {
        System.out.println("Enter the hdd of computer:");
        this.hdd = sc.nextLine();
    }
    public void setmobo() {
        System.out.println("Enter the mobo of computer:");
        this.mobo = sc.nextLine();
    }
    public String getname() {
        return this.name;
    }
    public String getcpu() {
        return this.processor;
    }
    public String gettype() {
        return this.type;
    }
    public String getram() {
        return this.ram;
    }
    public String gethdd() {
        return this.hdd;
    }
    public String getmobo() {
        return this.mobo;
    }
    public void display() {
        System.out.println("\tPC INFO");
        System.out.println("Name:" + getname() + "\nPROCESSOR:" + getcpu() +
"\nTYPE:" + gettype() + "\nRAM:" + getram() + "\nHARDDISk:" + gethdd() + "\nMOTHER
BOARD:" + getmobo());
    }
    public void update() {
        System.out.println("DO YOU WANT TO UPDATE ANY THING?");
        String choice = sc.nextLine().toLowerCase();
        if (choice.contentEquals("yes")) {
            System.out.println("WHAT DO YOU WANT TO
UPDATE?\n1.Name\n2.Processor\n3.Type\n4.RAM\n5.HardDisk\n6.MotherBoard");
            String option = sc.nextLine().toLowerCase();
            switch (option) {
                case "1", "name", "1.name", "1name" -> {
                    setname();
                    display();
                    update();
                }
                case "2", "processor", "2.processor", "2processor" -> {
                    setcpu();
                    display();
                    update();
                }
                case "3", "type", "3.type", "3type" -> {
                    settype();
                    display();
                    update();
                }
                case "4", "ram", "3.ram", "3ram" ->  {
                    setram();
                    display();
                    update();
                }
                case "5", "harddisk", "5.harddisk", "5harddisk" -> {
                    sethdd();
                    display();
                    update();
                }
                case "6", "motherboard", "6.motherboard", "6motherboard" -> {
                    setmobo();
                    display();
                    update();
                }
                default -> {
                    System.out.println("YOU ENTERED AN UNLISTED OPTION...");
                }
            }
        }
    }
}
```

**Output:**



```
Enter the name of computer:
Hp EliteBook
Enter the processor of computer:
i5
Enter the type of computer:
Laptop
Enter the ram of computer:
8Gb
Enter the hdd of computer:
256Gb
Enter the mobo of computer:
MotherBoard
        PC INFO
Name:Hp EliteBook
PROCESSOR:i5
TYPE:Laptop
RAM:8Gb
HARDDISk:256Gb
MOTHER BOARD:MotherBoard
DO YOU WANT TO UPDATE ANY THING?
no
```

**Task No 02:** Use Constructor to set the radius and height of cylinder and calculate surface area and Volume of cylinder.

**Code:**

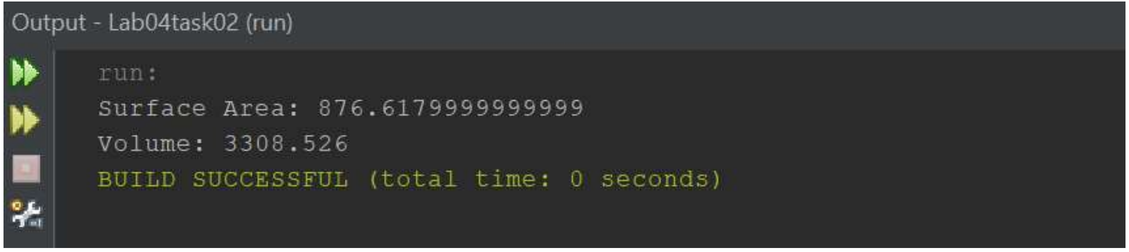
**Main:**

```
package lab04task02;

public class Lab04task02 {

    public static void main(String[] args) {
        Cylinder c1 = new Cylinder(9,13);

        System.out.print("Surface Area: ");
        System.out.println(c1.surfaceArea());
        System.out.print("Volume: ");
        System.out.println(c1.volume());
    }
}
```

**Cylinder:**

```
package lab04task02;

public class Cylinder {
    private int height;
    private int radius;

    Cylinder (int radius, int height){

        this.radius = radius;
        this.height = height;
    }
    public Double surfaceArea(){
        return 2*3.142*radius*radius+3.142*radius*height;
    }
    public Double volume(){
        return 3.142*radius*radius*height;
    }
}
```

**Output:**



```
run:
Surface Area: 876.6179999999999
Volume: 3308.526
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Task No 03:** Use constructor overloading to initialize a rectangle of length 4 and breadth 5 for using custom parameters.

## Code:

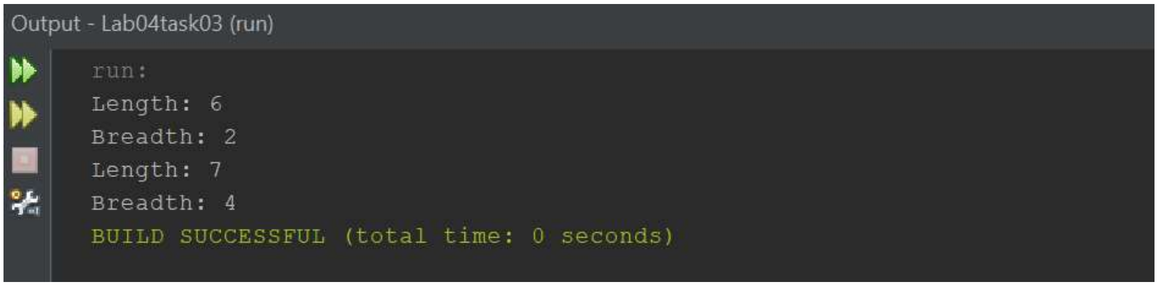
### Main:

```
package lab04task03;

public class Lab04task03 {

    public static void main(String[] args) {

        Rectangle c1 = new Rectangle ();
        Rectangle c2 = new Rectangle (7,4);

        c1.Display();
        c2.Display();
    }
}
```

### Employee:

```
package lab04task03;

public class Rectangle {

    private int length;
    private int breadth;

    Rectangle (){

        length = 6;
        breadth = 2;
    }
     Rectangle (int L, int b){

        length = L;
        breadth = b;
    }
    public void Display() {

        System.out.println("Length: " + length);
        System.out.println("Breadth: " + breadth);
    }
}
```

## Output:



```
run:
Length: 6
Breadth: 2
Length: 7
Breadth: 4
BUILD SUCCESSFUL (total time: 0 seconds)
```

**Task No 04:** Design then implement a class to represent a **Flight**. A Flight has a flight number, a source, a destination, and number of available seats. This should be implemented using proper access modifier. The class should have:

- **a.** A constructor to initialize the 4 instance variables. You must shorten the name of the source and the destination to 3 characters only if it is longer than 3 characters by a call to the method in the 'h' part.
- **b.** An overloaded constructor to initialize the flight number and the number of available seats instance variables only.
  (NOTE: Initialize the source and the destination instance variables to empty string, i.e.," ")
- **c.** An overloaded constructor to initialize the flight number instance variable only.
  (NOTE: Initialize the source and the destination instance variables to empty string; and the number of available seats to zero)
- **d.** A method **public void reserve (int numberOfSeats)** to reserve seats on the flight.
  (NOTE: You must check that there is enough number of seats to reserve)
- **e.** A method **public void cancel (int numberOfSeats)** to cancel one or more reservations.
- **f.** A **toString** method to easily return the flight information as follows:

```
Flight No: 1234

From: KAR

To: LAH

Available Seats: 18
```

- **g.** An **equal's** method to compare 2 flights.
  (NOTE: 2 Flights considered being equal if they have the same flight number)
- **h.** The following method:

```
private String shortAndCapital (String name) {

  if (name.length() <= 3) {

    return name.toUpperCase();

  } else {

    return name.substring(0,3).toUpperCase();

  }
```

Write a test class for the Flight class you wrote. You should try to use all the methods you wrote.

## Code:

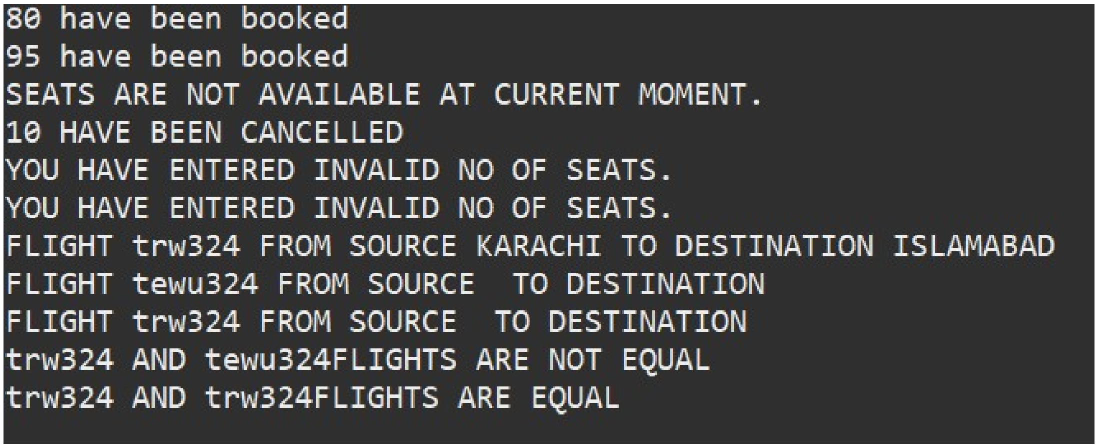
### Main:

```
package lab04task04;

public class Lab04task04 {

    public static void main(String[] args) {
        Flight f1 = new Flight("trw324", "KARACHI", "ISLAMABAD", 87);
        Flight f2 = new Flight("tewu324", 90);
        Flight f3 = new Flight("trw324");
        f1.reserve(80);
        f2.reserve(95);
        f3.reserve(67);
        f1.cancel(10);
        f2.cancel(30);
        f3.cancel(78);
        System.out.println(f1.tostring());
        System.out.println(f2.tostring());
        System.out.println(f3.tostring());
        boolean a = f1.equals(f2);
        if (a == true) {
            System.out.println(f1.fno + " AND " + f2.fno + "FLIGHTS ARE EQUAL");
        } else {
            System.out.println(f1.fno + " AND " + f2.fno + "FLIGHTS ARE NOT EQUAL");
        }
        boolean b = f1.equals(f3);
        if (b == true) {
            System.out.println(f1.fno + " AND " + f3.fno + "FLIGHTS ARE EQUAL");
        } else {
            System.out.println(f1.fno + " AND " + f3.fno + "FLIGHTS ARE NOT EQUAL");
        }
    }
}
```

**Flight:**

```
package lab04task04;

public class Flight {

    String fno, source, destination;
    int aseat;

    public Flight(String fno, String source, String destination, int aseat) {
        this.fno = fno;
        this.source = source;
        this.destination = destination;
        this.aseat = aseat;
    }
    public Flight(String fno, int aseat) {
        this.fno = fno;
        this.source = "";
        this.destination = "";
        this.aseat = aseat;
    }
    public Flight(String fno) {
        this.fno = fno;
        this.source = "";
        this.destination = "";
        this.aseat = 0;
    }
    public void reserve(int noofseats) {
        if (this.aseat > 0) {
            aseat = aseat - noofseats;
            System.out.println(noofseats + " have been booked");
        } else {
            System.out.println("SEATS ARE NOT AVAILABLE AT CURRENT MOMENT.");
        }
    }
    public void cancel(int noofseat) {
        if ((aseat > 0) && (aseat <= noofseat)) {
            aseat = -noofseat;
            System.out.println(noofseat + " HAVE BEEN CANCELLED");
        } else {
            System.out.println("YOU HAVE ENTERED INVALID NO OF SEATS.");
        }
    }
    public String tostring() {
        return ("FLIGHT " + this.fno + " FROM SOURCE " + this.source + " TO 
DESTINATION " + this.destination);
    }
    public boolean equals(Object obj) {
        if (obj == this) {
            return true;
        }
        if (!(obj instanceof Flight)) {
            return false;
        }
        Flight otherFlight = (Flight) obj;
        return (this.fno.equals(otherFlight.fno));
    }
    private String shortAndCapital(String name) {
        if (name.length() <= 3) {
            return name.toUpperCase();
        } else {
            return name.substring(0, 3).toUpperCase();
        }
    }
}
```

**Output:**



```
80 have been booked
95 have been booked
SEATS ARE NOT AVAILABLE AT CURRENT MOMENT.
10 HAVE BEEN CANCELLED
YOU HAVE ENTERED INVALID NO OF SEATS.
YOU HAVE ENTERED INVALID NO OF SEATS.
FLIGHT trw324 FROM SOURCE KARACHI TO DESTINATION ISLAMABAD
FLIGHT tewu324 FROM SOURCE  TO DESTINATION
FLIGHT trw324 FROM SOURCE  TO DESTINATION
trw324 AND tewu324FLIGHTS ARE NOT EQUAL
trw324 AND trw324FLIGHTS ARE EQUAL
```