Task 1: 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.

SOLUTION: MAIN METHOD:

```
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();    }}
```

Class:

```
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;}
```

OUTPUT:

```
Enter the name of computer:
Intel
Enter the processor of computer:
i5 10th gen
Enter the type of computer:
Laptop
Enter the ram of computer:
8gb ddr5
Enter the hdd of computer:
1 TB
Enter the mobo of computer:
Intel
        PC INFO
Name:Intel
PROCESSOR:i5 10th gen
TYPE:Laptop
RAM:8gb ddr5
HARDDISk:1 TB
MOTHER BOARD:Intel
DO YOU WANT TO UPDATE ANY THING?
yes
WHAT DO YOU WANT TO UPDATE?
1.Name
2.Processor
3.Type
4.RAM
5.HardDisk
6.MotherBoard
ram
```

```
    public void display(){    System.out.println("\tPC INFO");
System.out.println("Name:"+getname()+"\nPROCESSOR:"+getcpu()+"\nTYPE:"+gettype()+"\nRAM:"+getram()+"\n
HARDDISk:"+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":case"name":case"1.name":case"1name":{
       setname();        display();        update();        break;}
     case"2":case"processor":case"2.processor":case"2processor":{
       setcpu();        display();        update();        break;}
    case"3":case"type":case"3.type":case"3type":{
       settype();        display();        update();        break;}
    case"4":case"ram":case"3.ram":case"3ram":{
      setram();       display();       update();       break;}
    case"5":case"harddisk":case"5.harddisk":case"5harddisk":{
       sethdd();       display();       update();       break;}
case"6":case"motherboard":case"6.motherboard":case"6motherboard":{
       setmobo();       display();       update();       break;}
    default:{System.out.println("YOU ENTERED AN UNLISTED OPTION...");}}}}}
```

```
ram
Enter the ram of computer:
16gb ddr5
        PC INFO
Name:Intel
PROCESSOR:i5 10th gen
TYPE:Laptop
RAM:16gb ddr5
HARDDISk:1 TB
MOTHER BOARD:Intel
DO YOU WANT TO UPDATE ANY THING?
no
------------------------------------
BUILD SUCCESS
```

TASK 2: Use Constructor to set the radius and height of cylinder and calculate surface area and Volume of cylinder.

SOLUTION: MAIN METHOD:

```
   public static void main(String[] args) {
      Scanner in=new Scanner(System.in);
System.out.println("ENTER THE RADIUS OF CYLINDER:");
      double a=in.nextDouble();
System.out.println("ENTER THE HEIGHT OF CYLINDER:");
      double b=in.nextDouble();
      cylinder c=new cylinder(a,b);
```

OUTPUT:

```
run:
ENTER THE RADIUS OF CYLINDER:
5
ENTER THE HEIGHT OF CYLINDER:
5
SURFACE AREA OF CYLINDER IS 314.1592653589793
VOLUME OF CYLINDER: 392.69908169872417
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
    c.sa();      c.v();   }  }
```

CLASS:

```
import java.lang.Math;
public class cylinder {   double radius,height;
   public cylinder(double radius,double height)   {      this.radius=radius;   this.height=height;   }
   public void sa(){   double a1=2*Math.PI*this.radius*this.height;
   double a2=2*Math.PI*Math.pow(this.radius,2);
   double a=a1+a2;
   System.out.println("SURFACE AREA OF CYLINDER IS "+a);   }
   public void v(){   double v=Math.PI*Math.pow(this.radius,2)*height;
   System.out.println("VOLUME OF CYLINDER: " +v);   }  }
```

TASK 3: Use constructor overloading to initialize a rectangle of length 4 and breadth 5 for using custom parameters.

SOLUTION: **MAIN METHOD:**

```
   public static void main(String[] args) {
      Scanner in=new Scanner(System.in);
      System.out.println("DEFAULT CONSTRUCTOR:");
      rec r=new rec();      r.display();
      System.out.println("ENTER THE LENGTH:");
      double l=in.nextDouble();
      System.out.println("ENTER THE BREADTH:");
      double b=in.nextDouble();
      System.out.println("\n\nPARAMETERIZED CONSTRUCTORS:");
      rec r1=new rec(l,b);      r1.display(); }}
```

OUTPUT:

```
run:
DEFAULT CONSTRUCTOR:
LENGTH:4.0
BREADTH:5.0
ENTER THE LENGTH:
7
ENTER THE BREADTH:
9


PARAMETERIZED CONSTRUCTORS:
LENGTH:7.0
BREADTH:9.0
BUILD SUCCESSFUL (total time: 4 seconds)
```

CLASS:

```
public class rec {   double length,breadth;
    public rec(){   length=4;   breadth=5;   }
   public rec(double length,double breadth){   this.length=length;   this.breadth=breadth;   }
   public void display(){     System.out.println("LENGTH:"+length+"\nBREADTH:"+breadth);}}
```

Task 4: Design then implement a class to represent a **Flight**. A Flight has a *flight number*, a *source*, a *destination* and a *number of available seats.* The class should have:

a) A **constructor** to initialize the 4 instance variables. You have to 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 .
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 have to 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

MAIN METHOD:

```
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");} }}
```

Class:

```
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
------------------------------------------------------------
BUILD SUCCESS
```