1.Red code Technology

Casual Employee:

```
public class CasualEmployee extends Employee{

 private int supplementaryHours;
 private double foodAllowance;

 public int getSupplementaryHours() {
 return supplementaryHours;
 }
public void setSupplementaryHours(int supplementaryHours) {
this.supplementaryHours = supplementaryHours;
 }
 public double getFoodAllowance() {
 return foodAllowance;
 }
 public void setFoodAllowance(double foodAllowance) {
 this.foodAllowance = foodAllowance;
 }
 public CasualEmployee(String EmployeeId, String EmployeeName, int yearsOfExperience,
String gender, double salary, int supplementaryHours, double foodAllowance)
 {
super(EmployeeId, EmployeeName,yearsOfExperience,gender,salary);
 this.supplementaryHours=supplementaryHours;
 this.foodAllowance=foodAllowance;
 }

 public double calculateIncrementedSalary(int incrementPercentage)
{
```

```
double total =(supplementaryHours*1000)+foodAllowance+this.salary;
double incsalary=total+(total*incrementPercentage/100);
return incsalary;
}

}
```

Employee:

```
public abstract class Employee {

protected String EmployeeId;
protected String EmployeeName;
protected int yearsOfExperience;
protected String gender;
protected double salary;
public abstract double calculateIncrementedSalary(int incrementPercentage);

 public String getEmployeeId() {
 return EmployeeId;
 }
 public void setEmployeeId(String employeeId) {
 this.EmployeeId = employeeId;
 }
 public String getEmployeeName() {
 return EmployeeName;
 }
 public void setEmployeeName(String employeeName) {
 this.EmployeeName = employeeName;
 }
 public int getYearsOfExperience() {
```

```
 return yearsOfExperience;
 }
 public void setYearsOfExperience(int yearsOfExperience) {
 this.yearsOfExperience = yearsOfExperience;
 }
 public String getGender() {
 return gender;
 }
 public void setGender(String gender) {
 this.gender = gender;
}
 public double getSalary() {
 return salary;
 }
 public void setSalary(double salary) {
 this.salary = salary;
 }
 public Employee(String employeeId, String employeeName, int yearsOfExperience, String
gender, double salary) {
 super();
 this.EmployeeId = employeeId;
 this.EmployeeName = employeeName;
 this.yearsOfExperience = yearsOfExperience;
 this.gender = gender;
this.salary=salary;
}
 }
```

Permanent Employee:

```
public class PermanentEmployee extends Employee{
```

```
private double medicalAllowance;
private double VehicleAllowance;

 public double getMedicalAllowance() {
 return medicalAllowance;
 }

 public void setMedicalAllowance(double medicalAllowance) {
 this.medicalAllowance = medicalAllowance;
 }

 public double getVehicleAllowance() {
 return VehicleAllowance;
 }

 public void setVehicleAllowance(double vehicleAllowance) {
 VehicleAllowance = vehicleAllowance;
 }

 public PermanentEmployee(String EmployeeId, String EmployeeName, int
yearsOfExperience, String gender, double salary, double medicalAllowance, double
vehicleAllowance)
 {
 super(EmployeeId, EmployeeName, yearsOfExperience, gender, salary);
 this.medicalAllowance=medicalAllowance;
 this.VehicleAllowance=vehicleAllowance;
 }

 public double calculateIncrementedSalary(int incrementPercentage)
```

```
{
double total=medicalAllowance + VehicleAllowance+this.salary;
double incsalary=total+(total*incrementPercentage/100);
return incsalary;
}
}
```

Trainee Employees:

```
public class TraineeEmployees extends Employee{

private int supplementaryTrainingHours;
 private int scorePoints;

 public int getSupplementaryTrainingHours() {
 return supplementaryTrainingHours;
 }
 public void setSupplementaryTrainingHours(int supplementaryTrainingHours) {
 this.supplementaryTrainingHours = supplementaryTrainingHours;
 }
 public int getScorePoints() {
 return scorePoints;
 }
 public void setScorePoints(int scorePoints) {
 this.scorePoints = scorePoints;
 }

 public TraineeEmployees(String EmployeeId, String EmployeeName, int yearsOfExperience,
String gender, double salary, int supplementaryTrainingHours, int scorePoints)
 {
 super(EmployeeId, EmployeeName, yearsOfExperience, gender, salary);
```

```
this.supplementaryTrainingHours=supplementaryTrainingHours;
this.scorePoints=scorePoints;
}

public double calculateIncrementedSalary(int incrementPercentage){
double total=(supplementaryTrainingHours*500)+(scorePoints*50)+this.salary;
double incsalary=total+(total*incrementPercentage/100);
return incsalary;
}



}
```

User Interface:

```
import java.util.Scanner;
public class UserInterface {

public static void main(String[] args){

Scanner sc=new Scanner(System.in);
System.out.println("Enter Employee Id");
String EmployeeId = sc.next();
System.out.println("Enter Employee name");
String EmployeeName = sc.next();
System.out.println("Enter Experience in years");
int yearsOfExperience = sc.nextInt();
System.out.println("Enter Gender");
String gender = sc.next();
System.out.println("Enter Salary");
double salary=sc.nextDouble();
```

```
double incSalary=0;
if(yearsOfExperience>=1 && yearsOfExperience <= 5)
{
System.out.println("Enter Supplementary Training Hours");
int supplementaryTrainingHours = sc.nextInt();
System.out.println("Enter Score Points");
int scorePoints = sc.nextInt();
TraineeEmployees te=new TraineeEmployees(EmployeeId, EmployeeName,
yearsOfExperience, gender, salary, supplementaryTrainingHours, scorePoints);
incSalary=te.calculateIncrementedSalary(5);
System.out.println("Incremented Salary is "+incSalary);
}
else if(yearsOfExperience>=6 && yearsOfExperience <=10)
{
System.out.println("Enter Supplementary Hours");
int supplementaryHours = sc.nextInt();
System.out.println("Enter Food Allowance");
double foodAllowance = sc.nextDouble();
CasualEmployee ce=new CasualEmployee(EmployeeId, EmployeeName, yearsOfExperience,
gender, salary, supplementaryHours, foodAllowance);
incSalary = ce.calculateIncrementedSalary(12);
System.out.println("Incremented Salary is "+incSalary);
}
else if(yearsOfExperience>=10 && yearsOfExperience <=25)
{
System.out.println("Enter Medical Allowance");
double medicalAllowance = sc.nextDouble();
System.out.println("Enter Vehicle Allowance");
double vehicleAllowance = sc.nextDouble();
```

```
 PermanentEmployee pe = new PermanentEmployee(EmployeeId, EmployeeName,
yearsOfExperience, gender, salary, medicalAllowance, vehicleAllowance);
 incSalary=pe.calculateIncrementedSalary(12);
 System.out.println("Incremented Salary is "+incSalary);
 }
 else
 System.out.println("Provide valid Years of Experience");
 }


 }
```

2.Dominion Cinemas

Book Movie Ticket:

```
public class BookAMovieTicket {
 protected String ticketId;
 protected String customerName;
 protected long mobileNumber;
 protected String emailId;
 protected String movieName;
 public void setticketId( String ticketId){
 this.ticketId=ticketId;
 }
 public void setcustomerName( String customerName){
 this.customerName=customerName;
 }
 public void setmobileNumber( long mobileNumber){
 this.mobileNumber=mobileNumber;
 }
 public void setemailId( String emailId){
```

```
 this.emailId=emailId;
 }
 public void setmovieName( String movieName){
 this.movieName=movieName;
 }
 public String getticketId(){
 return ticketId;
 }
 public String getcustomerName(){
 return customerName;
 }
 public String getemailId(){
 return emailId;
 }
 public String getmovieName(){
 return movieName;
 }
 public long getmobileNumber(){
 return mobileNumber;
 }
 public BookAMovieTicket(String ticketId,String customerName,long mobileNumber,String
emailId,String movieName)
{
this.ticketId=ticketId;
this.customerName=customerName;
this.mobileNumber=mobileNumber;
this.emailId=emailId;
this.movieName=movieName;
}
```

```
}
```

Gold Ticket:

```
public class GoldTicket extends BookAMovieTicket {
public GoldTicket(String ticketId, String customerName, long mobileNumber,
String emailId, String movieName) {
super(ticketId, customerName, mobileNumber, emailId, movieName);
}
public boolean validateTicketId(){
int count=0;
if(ticketId.contains("GOLD"));
count++;
char[] cha=ticketId.toCharArray();
for(int i=4;i<7;i++){
if(cha[i]>='1'&& cha[i]<='9')
count++;
}
if(count==4)
return true;
else
return false;
}
public double calculateTicketCost(int numberOfTickets,String ACFacility){
double amount;
if(ACFacility.equals("yes")){
amount=500*numberOfTickets;
}
else{
amount=350*numberOfTickets;
```

```
}
return amount;
}
}
```

Platinum Ticket:

```
public class PlatinumTicket extends BookAMovieTicket
 {
public PlatinumTicket(String ticketId, String customerName, long mobileNumber,String
emailId, String movieName)
{
super(ticketId, customerName, mobileNumber, emailId, movieName);
}
public boolean validateTicketId(){
int count=0;
if(ticketId.contains("PLATINUM"));
count++;
char[] cha=ticketId.toCharArray();
for(int i=8;i<11;i++){
if(cha[i]>='1'&& cha[i]<='9')
count++;
}
if(count==4)
return true;
else
return false;
}
public double calculateTicketCost(int numberOfTickets,String ACFacility){
double amount;
if(ACFacility.equals("yes")){
```

```
amount=750*numberOfTickets;
}
else{
amount=600*numberOfTickets;
}
return amount;
}
}
```

Silver Ticket:

```
public class SilverTicket extends BookAMovieTicket{
public SilverTicket(String ticketId, String customerName, long mobileNumber,String emailId, String movieName)
{
super(ticketId, customerName, mobileNumber, emailId, movieName);
}
public boolean validateTicketId(){
int count=0;
if(ticketId.contains("SILVER"));
count++;
char[] cha=ticketId.toCharArray();
for(int i=6;i<9;i++){
if(cha[i]>='1'&& cha[i]<='9')
count++;
}
if(count==4)
return true;
else
return false;
}
```

```
public double calculateTicketCost(int numberOfTickets,String ACFacility){
double amount;
if(ACFacility.equals("yes")){
amount=250*numberOfTickets;
}
else{
amount=100*numberOfTickets;
}
return amount;
}
}
```

User Interface:

```
import java.util.*;
public class UserInterface {
public static void main(String[] args) {
Scanner sc=new Scanner(System.in);
System.out.println("Enter Ticket Id");
String tid=sc.next();
System.out.println("Enter Customer Name");
String cnm=sc.next();
System.out.println("Enter Mobile Number");
long mno=sc.nextLong();
System.out.println("Enter Email id");
String email=sc.next();
System.out.println("Enter Movie Name");
String mnm=sc.next();
System.out.println("Enter number of tickets");
int tno=sc.nextInt();
```

```
System.out.println("Do you want AC or not");
String choice =sc.next();
if(tid.contains("PLATINUM")){
PlatinumTicket PT=new PlatinumTicket(tid,cnm,mno,email,mnm);
boolean b1=PT.validateTicketId();
if(b1==true){
double cost =PT.calculateTicketCost(tno, choice);
System.out.println("Ticket cost is "+ cost);
}
else if(b1==false){
System.out.println("Provide valid Ticket Id");
System.exit(0);
}
}
else if(tid.contains("GOLD")){
GoldTicket GT=new GoldTicket(tid,cnm,mno,email,mnm);
boolean b2=GT.validateTicketId();
if(b2==true){
double cost=GT.calculateTicketCost(tno, choice);
System.out.println("Ticket cost is "+cost);
}
else if (b2==false){
System.out.println("Provide valid Ticket Id");
System.exit(0);
}
}
else if(tid.contains("SILVER")){
SilverTicket ST=new SilverTicket(tid,cnm,mno,email,mnm);
boolean b3=ST.validateTicketId();
```

```
if(b3==true){
double cost=ST.calculateTicketCost(tno, choice);
System.out.println("Ticket cost is "+cost);
}
else if(b3==false){
System.out.println("Provide valid Ticket Id");
System.exit(0);
}
}
}
}
```

3.Little Innovators

Main:

```
import java.util.*;
public class Main {

    public static void main(String args[])
                                        {
                                            System.out.println("Enter the String: ");
                                            Scanner sc=new Scanner(System.in);
                                            String st=sc.nextLine();
                                            String op=st;
                                            st=st.replaceAll(" ","");
                                            boolean d=st.matches("[a-zA-Z]+");
                                            if(!d)
                                            {
                                               System.out.println("Invalid Slogan");
                                            }
```

```
                                        else
                                        {
                                                char a[]=st.toCharArray();
                                                char b[]=new char[100];
                                                b[0]='0';
                                                int same=0,i=a.length,j=0,l=0;
                                                while(j<i)
        {
           int count=0;
           for(int k=0;k<i;k++)
           {
              if(a[j]==a[k])
              {
                 count++;
              }
           }
           if(count==1)
           {
              l++;
              b[l]=a[j];
              j++;
           }
           else
           {
              j++;
           }
        }
                                                if(l==(i-l))
                                                        System.out.println("All the guidelines
are satisfied for "+op);
```

```
                                        else
                                            System.out.println(op+" does not satisfy
the guideline");
                                    }
                                }
}
```

4.Kidsor Home appliances

Air Conditioner:

```
public class AirConditioner extends ElectronicProducts {
 private String airConditionerType;
 private double capacity;
 public AirConditioner(String productId, String productName, String batchId, String
dispatchDate, int warrantyYears, String airConditionerType, double capacity) {
 super(productId, productName, batchId, dispatchDate, warrantyYears);
 this.airConditionerType = airConditionerType;
 this.capacity = capacity;
 }
 public String getAirConditionerType() {
 return airConditionerType;
 }
 public void setAirConditionerType(String airConditionerType) {
 this.airConditionerType = airConditionerType;
 }
 public double getCapacity() {
 return capacity;
 }
 public void setCapacity(double capacity) {
```

```
this.capacity = capacity;
}
public double calculateProductPrice(){
double price = 0;
if(airConditionerType.equalsIgnoreCase("Residential")){
if (capacity == 2.5){
price = 32000;
}
else if(capacity == 4){
price = 40000;
}
else if(capacity == 5.5){
price = 47000;
}
}
else if(airConditionerType.equalsIgnoreCase("Commercial")){
if (capacity == 2.5){
price = 40000;
}
else if(capacity == 4){
price = 55000;
}
else if(capacity == 5.5){
price = 67000;
}
}
else if(airConditionerType.equalsIgnoreCase("Industrial")){
if (capacity == 2.5){
price = 47000;
```

```
 }
 else if(capacity == 4){
 price = 60000;
 }
 else if(capacity == 5.5){
 price = 70000;
 }
 }
 return price;
 }
}
```

Electronic Products:

```
public class ElectronicProducts {
 protected String productId;
 protected String productName;
 protected String batchId;
 protected String dispatchDate;
 protected int warrantyYears;
 public ElectronicProducts(String productId, String productName, String batchId,
String dispatchDate, int warrantyYears) {
 this.productId = productId;
 this.productName = productName;
 this.batchId = batchId;
 this.dispatchDate = dispatchDate;
 this.warrantyYears = warrantyYears;
 }
 public String getProductId() {
 return productId;
```

```
}
public void setProductId(String productId) {
this.productId = productId;
}
public String getProductName() {
return productName;
}
public void setProductName(String productName) {
this.productName = productName;
}
public String getBatchId() {
return batchId;
}
public void setBatchId(String batchId) {
this.batchId = batchId;
}
public String getDispatchDate() {
return dispatchDate;
}
public void setDispatchDate(String dispatchDate) {
this.dispatchDate = dispatchDate;
}
public int getWarrantyYears() {
return warrantyYears;
}
public void setWarrantyYears(int warrantyYears) {
this.warrantyYears = warrantyYears;
}
}
```

LED TV:

```
public class LEDTV extends ElectronicProducts {
 private int size;
 private String quality;
 public LEDTV(String productId, String productName, String batchId, String
dispatchDate, int warrantyYears, int size, String quality) {
 super(productId, productName, batchId, dispatchDate, warrantyYears);
 this.size = size;
 this.quality = quality;
 }
 public int getSize() {
 return size;
 }
 public void setSize(int size) {
 this.size = size;
 }
 public String getQuality() {
 return quality;
 }
 public void setQuality(String quality) {
 this.quality = quality;
 }
 public double calculateProductPrice(){
 double price = 0;
 if(quality.equalsIgnoreCase("Low")){
 price = size * 850;
 }
 else if(quality.equalsIgnoreCase("Medium")){
 price = size * 1250;
```

```
 }
 else if(quality.equalsIgnoreCase("High")){
 price = size * 1550;
 }
 return price;
 }
}
```

Microwave Oven:

```
public class MicrowaveOven extends ElectronicProducts{
 private int quantity;
 private String quality;
 public MicrowaveOven(String productId, String productName, String batchId, String
dispatchDate, int warrantyYears, int quantity, String quality) {
 super(productId, productName, batchId, dispatchDate, warrantyYears);
 this.quantity = quantity;
 this.quality = quality;
 }
 public int getQuantity() {
 return quantity;
 }
 public void setQuantity(int quantity) {
 this.quantity = quantity;
 }
 public String getQuality() {
 return quality;
 }
 public void setQuality(String quality) {
 this.quality = quality;
```

```
 }
 public double calculateProductPrice(){
 double price = 0;
 if(quality.equalsIgnoreCase("Low")){
 price = quantity * 1250;
 }
 else if(quality.equalsIgnoreCase("Medium")){
 price = quantity * 1750;
 }
 else if(quality.equalsIgnoreCase("High")){
 price = quantity * 2000;
 }
 return price;
 }
}
```

User Interface:

```
import java.util.Scanner;
public class UserInterface {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Product Id");
 String productId = sc.next();
 System.out.println("Enter Product Name");
 String productName = sc.next();
 System.out.println("Enter Batch Id");
 String batchId = sc.next();
 System.out.println("Enter Dispatch Date");
```

```
String dispatchDate = sc.next();
System.out.println("Enter Warranty Years");
int warrantyYears = sc.nextInt();
double price;
String quality;
switch(productName){
case "AirConditioner":
System.out.println("Enter type of Air Conditioner");
String type = sc.next();
System.out.println("Enter quantity");
double capacity = sc.nextDouble();
AirConditioner ac = new AirConditioner(productId, productName, batchId,
dispatchDate, warrantyYears, type, capacity);
price = ac.calculateProductPrice();
System.out.printf("Price of the product is %.2f", price);
break;
case "LEDTV":
System.out.println("Enter size in inches");
int size = sc.nextInt();
System.out.println("Enter quality");
quality = sc.next();
LEDTV l = new LEDTV(productId, productName, batchId, dispatchDate,
warrantyYears, size, quality);
price = l.calculateProductPrice();
System.out.printf("Price of the product is %.2f", price);
break;
case "MicrowaveOven":
System.out.println("Enter quantity");
int quantity = sc.nextInt();
```

```
System.out.println("Enter quality");
quality = sc.next();
MicrowaveOven m = new MicrowaveOven(productId, productName, batchId,
dispatchDate, warrantyYears, quantity, quality);
price = m.calculateProductPrice();
System.out.printf("Price of the product is %.2f", price);
break;
default:
System.out.println("Provide a valid Product name");
System.exit(0);
}
}
}
```

5. Reverse a word

```
import java.util.*;

class HelloWorld {
    public static void main(String[] args) {
String[] words ;
Scanner myObj = new Scanner(System.in);

String sentence= myObj.nextLine();
words=sentence.split(" ");
if(words.length<3)
System.out.println("Invalid Sentence");
else{
```

```
String a=words[0].substring(0,1);
String b=words[1].substring(0,1);
String c=words[2].substring(0,1);
if(a.equalsIgnoreCase(b)&&b.equalsIgnoreCase(c))
{
StringBuilder input1 = new StringBuilder();

      input1.append(words[words.length-1]);
    // reverse StringBuilder input1

     input1= input1.reverse();
     input1.append(words[0]);

System.out.println(input1);
}
else {

StringBuilder input1 = new StringBuilder();

      input1.append(words[0]);
    // reverse StringBuilder input1

     input1= input1.reverse();
     input1.append(words[words.length-1]);

System.out.println(input1);
}

}
```

```
    }
}
```