Name: Vince Hendrick Manalad

Subject: Programming **Instructure: Sir RM**

// Chapter 1

Case Problem No 1.0: Carly's Catering provides meals for parties and special events. Write a program that displays Carly's motto, which is "Carly's makes the food that makes it a party." Save the file as CarlysMotto.java

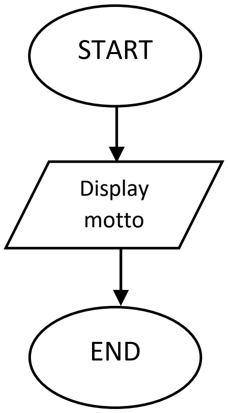
Problem:

Display Carly's Motto

Pseudocode:

```
1.Start
2. Display the motto which is," Carly's makes the food that makes party."
3. End
```

Flowcharting:



Coding:

```
public class CarlysMotto {
   public static void main(String [] args){
System.out.println("Caly's makes the food that make it a party.");
   }
}
```

Programm Output:

```
Carly's makes the food that makes it a party
```

Case Problem No 2.0: Sammy's Seashore Supplies rents beach equipment such as kayaks, canoes, beach chairs, and umbrellas to tourists. Write a program that displays Sammy's motto, which is "Sammy's makes it fun in the sun." Save the file as SammysMotto.java

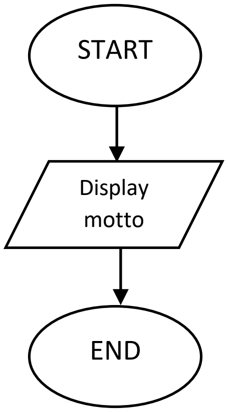
Problem:

Display Sammy's Motto

Pseudocode:

```
1.Start
2.Display the motto which is, "Sammy's makes it fun in the sun."
3.End
```

Flowcharting:



Code:

```
public class SammysMotto {
   public static void main(String [] args){
System.out.println("Sammy's make it funin the sun ");
   }
}
```

Program Output:

```
Sammy's make it funin the sun
```

Case Problem No 1.1 Create a second program that displays the motto surrounded by a border composed of asterisks. Save the file as CarlysMotto2.java.

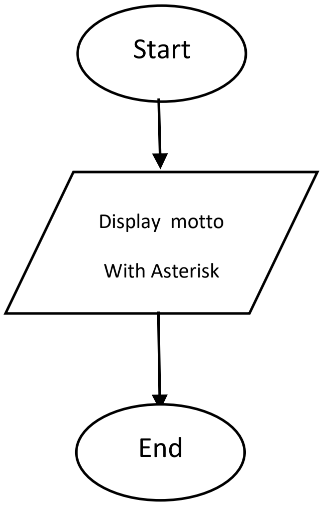
Problem:

Display Carly’s motto with Border

Pseudo Code:

```
1.Start
2.Display Carly’s Motto (“Carly's makes the food that make it a party”) with bordered Asterisk
3.End
```

Flowcharting:



Coding:

```
public class CalysMotto1 {
   public static void main(String []args){
      System.out.println("************************************************");
System.out.println("*        Carly's makes the food that make it a party.        *");
      System.out.println("************************************************");
   }
}
```

Program Output:

```
***********************************************
*          Carly's makes the food that make it a party.          *
***********************************************
```

Case Problem No 2.1 Create a second program that displays the motto surrounded by a border composed of repeated Ss. Save the file as SammysMotto2.java.

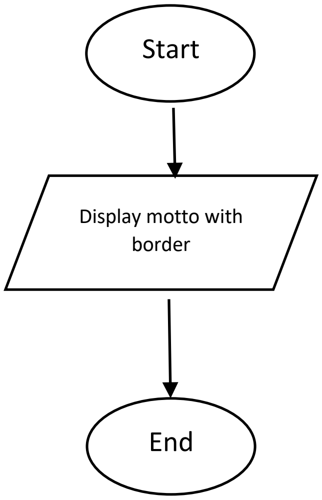
Problem:

Display Sammy’s Motto with border

Pseudo Code:

```
1.Start
2.Display Sammy’s motto (“Sammy's make it fun in the sun”) with bordered repeated S
3.End
```

Flowcharting:



Coding:

```
public class SammysMotto2 {
   public static void main(String [] args){
System.out.println("SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS");
System.out.println("S  Sammy's make it fun in the sun   S");
System.out.println("SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS");

   }

}
```

Program Output:

```
SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS

S  Sammy's make it fun in the sun   S

SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS
```

// Chapter 2

Case Problem No 1.2 Carly’s Catering provides meals for parties and special events. Write a program that prompts the user for the number of guests attending an event and then computes the total price, which is $35 per person. Display the company motto with the border that you created in the CarlysMotto2 class in Chapter 1, and then display the number of guests, price per guest, and total price. Also display a message that indicates true or false depending on whether the job is classified as a large event— an event with 50 or more guests. Save the file as CarlysEventPrice.java.

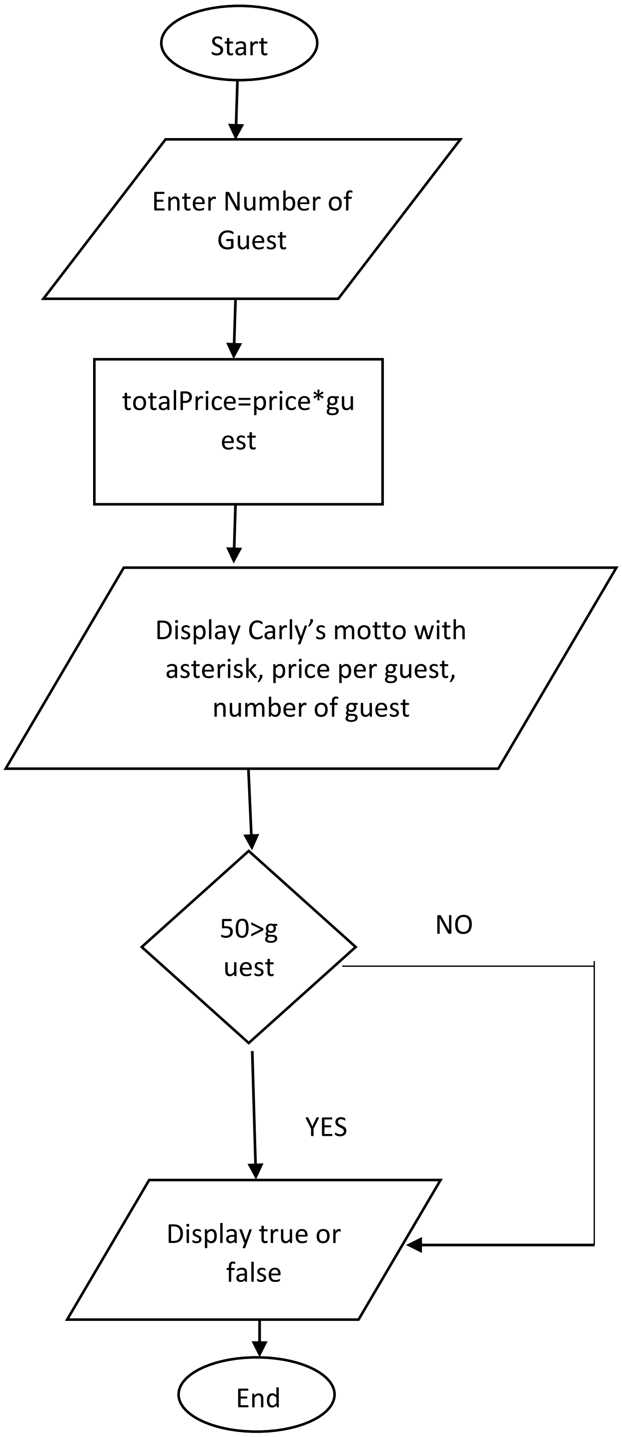
Problem:

Ask the user to enter number of guest and calculate the total price. And, display Carly’s motto together with the price per guest, number of guest and the total price also display if the party is big or not

Pseudocode:

```
1.Start

2.Enter Number of guest

3.Calculate the total price

4.Display the Carly’s motto together with the price per guest, number of guest, total price and if the event is big or not.

5.End
```

Flowcharting:



Coding:

```
import java.util.Scanner;

public class CalysEventPrice {
   public static void main(String[]args){
      final int price=35;
      Scanner input = new Scanner(System.in);
System.out.println("Enter number of guess");
int guess=input.nextInt();

inttotalprice=guess*price;

System.out.println("");
      System.out.println("*********************************************");
System.out.println("*Caly's makes the food that make it a party.*");
      System.out.println("*********************************************");

System.out.println("");
System.out.println("The number of guess are "+guess);
System.out.println("Price per guess "+price);
System.out.println("The total price is "+totalprice);
System.out.println("The job is classified as a large event? "+(guess>=50));
   }
}
```

Program Output:

```
Enter number of guess
65
*********************************************
*Caly's makes the food that make it a party.*
*********************************************
```

```
The number of guess are 65

Price per guess 35

The total price is 2275

The job is classified as a large event? True
```

Case Problem No 2.2 Sammy’s Seashore Supplies rents beach equipment such as kayaks, canoes, beach chairs, and umbrellas to tourists. Write a program that prompts the user for the number of minutes he rented a piece of sports equipment. Compute the rental cost as $40 per hour plus $1 per additional minute. (You might have surmised already that this rate has a logical flaw, but for now, calculate rates as described here. You can fix the problem after you read the chapter on decision making.) Display Sammy’s motto with the border that you created in the SammysMotto2 class in Chapter 1. Then display the hours, minutes, and total price. Save the file as SammysRentalPrice.java.

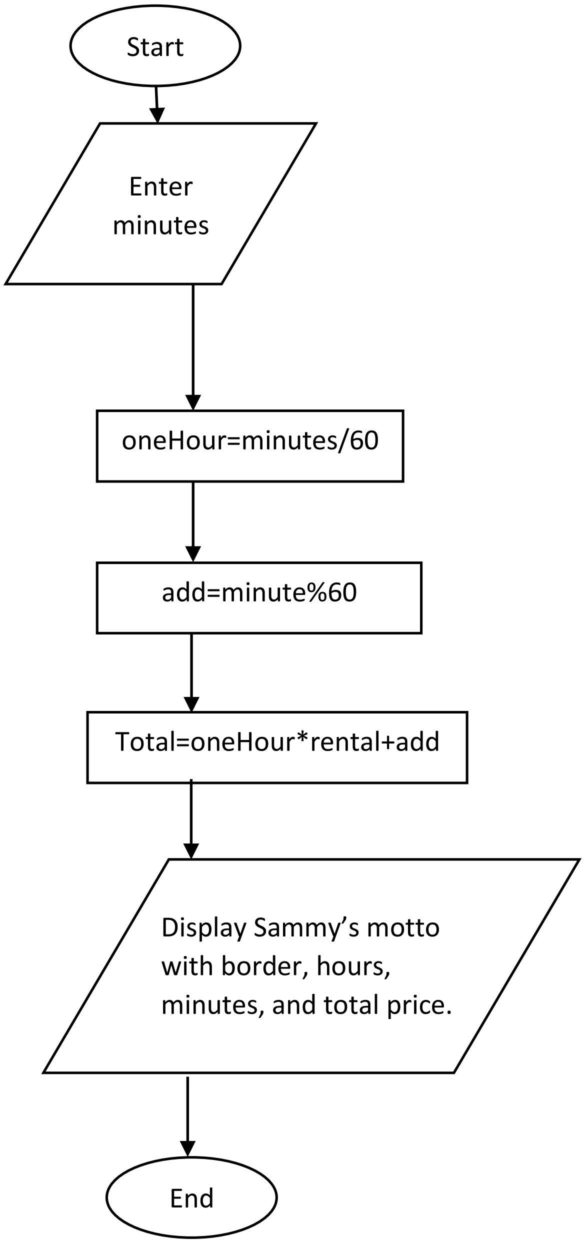
Problem:

Ask the user to enter the number of minute then compute the rental cost per hour and add $1 for every additional minute. Display the Sammy’s motto with border, hours, minutes and the total price.

Pseudocode:

```
1.Start
2.Enter number of minutes
3.Compute the rental cost
4.Add $1 for every additional minute
5.Display the Sammy’s motto with border, hours, minutes, and the total price.
6.End
```

Flowcharting:



Code:

```
import java.util.Scanner;

public class SammysRentalPrice {

  public static void main(String[]args){

    double rental=40;
```

```
        double perMinute=1;

        Scanner input=new Scanner(System.in);

System.out.println("Enter number of minutes ");
        double a=input.nextInt();

intoneHour=(int) (a/60);

double add=a%60;

        double total=oneHour*rental+add;

System.out.println("");
System.out.println("SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS");
System.out.println("S                                 S");
System.out.println("S  Sammy's make it fun in the sun S");
System.out.println("S                                 S");
System.out.println("SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS");
System.out.println("");

System.out.println("The rental price is "+rental);
System.out.println("The hour is "+oneHour);
System.out.println("The additional minute is "+add+" minute");
System.out.println("The total price are "+total);

    }
```

```
}
```

Program Output:

```
Enter number of minutes

60


SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS

S                                   S

S  Sammy's make it fun in the sun S

S                                   S

SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS


The rental price is 40.0

The hour is 1

The additional minute is 0 minute

The total price are 40
```

// Chapter 3

Case Problem No1.3 a Carly’s Cate.ring provides meals for parties and special events. In Chapter 2, you wrote an application that prompts the user for the number of guests attending an event, displays the company motto with a border, and then displays the price of the event and whether the event is a large one. Now modify the program so that the main() method contains only three executable statements that each call a method as follows:

The first executable statement calls a public static int method that prompts the user for the number of guests and returns the value to the main() method.

The second executable statement calls a public static void method that displays the company motto with the border.

The last executable statement passes the number of guests to a public static void method that computes the price of the event, displays the price, and displays whether the event is a large event. Save the file as CarlysEventPriceWithMethods.java

Problem:

Create a method that prompt the user and get the number of guest, create another method that display a Carly’s motto with border, and the last method is to compute and display the total price a price per guest and if the event is big or not.

Pseudocode:

```
1.Start
2. Create a method that prompt the user to get the number of guest.
3. Create another method that display a Carly’s motto with border.
4. The last method is to compute and display the total price a price per guest and if the event is big or not.
5.End
```

Code:

```
import java.util.Scanner;
public class CarlysEventPriceWithMethods {
  public static void main(String [] args){
  int one = 0, a;


    a = getFirst(one);
    second();
    third(a);


  }


  public static int getFirst(int one){
    }


        Scanner input=new Scanner(System.in);


System.out.println("Enter number of guest ");
        one=input.nextInt();


        return one;


    }


    public static void second(){
System.out.println("\n");
        System.out.println("*********************************************");
System.out.println("*Caly's makes the food that make it a party.*");
        System.out.println("*********************************************");
System.out.println("\n");
    }


    public static void third(int a){
int price=35, totalprice;


totalprice = a*price;


System.out.println("The number of guest are "+a);
System.out.println("Price per guest "+price);
System.out.println("The total price is "+totalprice);
System.out.println("The job is classified as a large event? "+(a>=50));
    }
```

```
}
```

Program Output:

Case Problem No1.3 b Create a class to hold Event data for Carly's Catering. The class contains:

Two public final static fields that hold the price per guest ($35) and the cutoff value for a large event (50 guests)

Three private fields that hold an event number, number of guests for the event, and the price. The event number is stored as a String because Carly plans to assign event numbers such as M312.

Two public set methods that set the event number (setEventNumber()) and the number of guests (setGuests()). The price does not have a set method because the setGuests() method will calculate the price as the number of guests multiplied by the price per guest every time the number of guests is set.

Three public get methods that return the values in the three nonstatic fields Save the file as Event.java.

Problem:

Pseudocode:

Flowcharting:

Code:

```
public class Event {



   // two public final static


   public final static int PRICE_PER_GUEST = 35;

   public final static int CUTT_OFF_VALUE = 50;
   // three private fields


   private String eventNumber;
   private int numberOfGuest;
   private int price;

   // two public set static set of method


   public void setEventNumber(String eventNum){
eventNumber=eventNum;
   }

   public void setNumberOfGuest(int guestNum){
numberOfGuest=guestNum;
      price = numberOfGuest*PRICE_PER_GUEST;
   }



   // three public get method


   public String getEventNumber(){


      return eventNumber;
   }

   public int getNumberOfGuest(){
        return numberOfGuest;
    }

    public int getPrice(){

        return price;
    }

}
```

Program Output:

Case Problem No1.3 c Use the CarlysEventPriceWithMethods class you created in Step 1a as a starting point for a program that demonstrates the Event class you created in Step 1b, but make the following changes:

You already have a method that gets a number of guests from a user; now add a method that gets an event number. The main() method should declare an Event object, call the two data entry methods, and use their returned values to set the fields in the Event object.

Call the method from the CarlysEventPriceWithMethods class that displays the company motto with the border. The method is accessible because it is public, but you must fully qualify the name because it is in another class.

Revise the method that displays the event details so that it accepts the newly created Event object. The method should display the event number, and it should still display the number of guests, the price per guest, the total price, and whether the event is a large event. Save the program as EventDemo.java.

Code:

```
import java.util.Scanner;
public class EventDemo {
    public static void main (String [] args){
```

```
    Event eve = new Event();

    eve.setEventNumber(eventNum());

    eve.setNumberOfGuest(first());

    CarlysEventWithMethod.second();


    compute(eve.getEventNumber(),eve.getNumberOfGuest(), eve.getPrice(),
eve.PRICE_PER_GUEST, eve.CUTT_OFF_VALUE);

  }

  public static int first(){

    Scanner inp= new Scanner(System.in);


    System.out.println("Enter the Number of Guests  >");

    int a = inp.nextInt();


    return a;

  }

  public static String eventNum(){

    Scanner inp = new Scanner(System.in);


    System.out.println("Enter the event number >");

    String a = inp.nextLine();


    return a;

  }

      public static void compute(String a, int b, int c, int d, int e){


    System.out.println("event number >"+a);
```

```
        System.out.println("number of guest >"+b);

        System.out.println("price per guest >"+d);

        System.out.println("total price >"+c);

        System.out.println("big event >"+(b>=e));

    }

}
```

Program Output:

```
Enter the event number >

Ma45

Enter the Number of Guests  >

45


************************************************

* Carly’s makes the food that makes it a party *

************************************************


event number >Ma45

number of guest >45

price per guest >35

total price >1575

big event >false
```

Case Problem No 2.3 a Sammy’s Seashore Supplies rents beach equipment such as kayaks, canoes, beach chairs, and umbrellas to tourists. In Chapter 2, you wrote an application that prompts the user for the number of minutes a piece of sports equipment was rented, displays the company motto with a border, and displays the price for the rental. Now modify the program so that the main() method contains only three executable statements that each call a method as follows:

```
   }
```

The first executable statement calls a method that prompts the user for the rental time in minutes and returns the value to the main() method.

The second executable statement calls a method that displays the company motto with the border.

The last executable statement passes the number of minutes to a method that computes the hours, extra minutes, and price for the rental, and then displays all the details. Save the file as SammysRentalPriceWithMethods.java

Code:

```
import java.util.Scanner;

public class SammysRentalPriceWithMethod {

   public static void main(String [] args){

int one = 0,a;


      a=first(one);

two();

      three(a);


   }


   public static int first(int one){


      Scanner input=new Scanner (System.in);

System.out.println("Enter number of minute ");

      one = input.nextInt();



      return one;

   }
```

```
    public static void two(){
System.out.println("");
System.out.println("SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS");
System.out.println("S                              S");
System.out.println("S  Sammy's make it fun in the sun S");
System.out.println("S                              S");
System.out.println("SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS");
System.out.println("");
    }

    public static void three(int a){

intoneHour=(int) (a/60);

int add=a%60;

int total=(int) (oneHour*a+add);

System.out.println("The rental price is "+a);
System.out.println("The hour is "+oneHour);
System.out.println("The additional minute is "+add+" minute");
System.out.println("The total price are "+total);
    }

}
```

Program Output:

```
Enter number of minute

60


SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS

S                         S

S  Sammy's make it fun in the sun S

S                         S

SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS


The rental price is 60

The hour is 1

The additional minute is 0 minute

The total price are 60
```

Case Problem No 2.3 b Create a class to hold Rental data for Sammy’s Seashore Supplies. The class contains:

Two public final static fields that hold the number of minutes in an hour and the hourly rental rate ($40)

Four private fields that hold a contract number, number of hours for the rental, number of minutes over an hour, and the price. The contract number is stored as a String because Sammy plans to assign contract numbers such as K681.

Two public set methods. One sets the contract number (setContractNumber()). The other is named setHoursAndMinutes(), and it accepts the number of minutes for the rental and then sets the hours, extra minutes over an hour, and the total price. Recall from Chapter 2 that the price is $40 per hour plus $1 for every extra minute.

Four public get methods that return the values in the four nonstatic fields Save the file as Rental.java.

Code:

```
public class Rental {

   public static int NUMBER_OF_MINUTE_IN_HOUR = 60;
   public static int HOURLY_RENTAL_RATE= 40;

   private String contactNum;
   private int numberOfHours;
   private int minutesOverHour;
   private int price;
   public void setContactNumber(String a){
      contactNum=a;
   }

   public void setHoursAndMinutes(int b){
      numberOfHours = b/NUMBER_OF_MINUTE_IN_HOUR;   // oneHour
      minutesOverHour = b%NUMBER_OF_MINUTE_IN_HOUR;          //aditional minute
      price = numberOfHours*HOURLY_RENTAL_RATE+minutesOverHour;   //total

   }

   public String getContactNumber(){
      return contactNum;
   }

   public int getNumberOfHours(){
      return numberOfHours;
   }

   public int getMinutesOverHour(){
      return minutesOverHour;
```

```
    }

    public int getPrice(){
        return price;
    }
}
```

Program Output:

// No output

Case Problem No 2.3 c Use the SammysRentalPriceWithMethods class you created in Step 2a as a starting point for a program that demonstrates the Rental class you created in Step 2b, but make the following changes:

You already have a method that gets a number of minutes from a user; now add a method that gets a contract number. The main() method should declare a Rental object, call the two data entry methods, and use their returned values to set the fields in the Rental object.

From the SammysRentalPriceWithMethods class, call the RentalDemo method that displays the company motto with the border. The method is accessible because it is public, but you must fully qualify the name because it is in another class.

Revise the method that displays the rental details so that it accepts the newly created Rental object. The method should display the contract number, and it should still display the hours and minutes, the hourly rate, and the total price. Save the program as RentalDemo.java

Code:

```
import java.util.Scanner;
public class RentalDemo {
    public static void main (String [] args){
        Rental rent = new Rental();
        rent.setContactNumber(contactMethod());
    rent.setHoursAndMinutes(minutesMethod());

    SammysRentalPriceWithMethod.two();



displayMethod(rent.getContactNumber(),rent.getMinutesOverHour(),rent.getNumberOfHours(
),rent.getPrice(),
        rent.NUMBER_OF_MINUTE_IN_HOUR,rent.HOURLY_RENTAL_RATE);

  }

  public static int minutesMethod(){
    Scanner inp = new Scanner(System.in);

    System.out.println("Enter number of minutes > ");
    int a = inp.nextInt();

    return a;
  }

  public static String contactMethod(){
    Scanner inp = new Scanner(System.in);

    System.out.println("Enter contact number > ");
    String a = inp.nextLine();

    return a;
    }

    public static void displayMethod(String a, int b, int c, int d, int e, int f){

        System.out.println("Contact Number >"+a);
        System.out.println("Number of Hours >"+c);
        System.out.println("Number of minutes  >"+b);
        System.out.println("The Hourly Rate  >"+f);
        System.out.println("The total price are  >"+d);
    }
}
```

Program Output:

```
Enter contact number >
K681
Enter number of minutes >
65

SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS
S                                 S
S  Sammy's make it fun in the sun S
S                                 S
SsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsSsS

Contact Number >K681
```

```
Number of Hours >1

Number of minutes  >5

The Hourly Rate  >40

The total price are  >45
```