slip4

```
class s4q1{

public static void main(String args[])

{

 String s1="hello world";

 System.out.println("alternate string :");

 prints4q1(s1);

}

public static void prints4q1(String str)

{

for(int i=0;i<str.length();i+=2)

 {

  System.out.println(str.charAt(i)+" ");

}

System.out.println();

}

}
```

```
import java.awt.*;

import java.awt.event.*;


class ArithmeticCalculatorApplet  extends java.applet.Applet implements
ActionListener{

Label n1L,n2L,resultL;

TextField n1F,n2F,resultF;

Button addB,subB,divB,mulB;
```

```
public void init()
{
 setLayout(new FlowLayout());
 n1L=new Label("number 1:");
 add(n1L);
 n1F= new TextField(10);
 add(n1F);

 n2L = new Label("number 2:");
 add(n1L);
 n2F= new TextField(10);
 add(n2F);

 addB = new Button("+");
 add(addB);
 subB= new Button("-");
 add(subB);
 divB= new Button("/");
 add(divB);
 mulB= new Button("*");
 add(mulB);

 addB.addActionListener(this);
 subB.addActionListener(this);
 divB.addActionListener(this);
 mulB.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
double n1,n2,result;
n1=Double.parseDouble(n1F.getText());
n2=Double.parseDouble(n2F.getText());

if(e.getSource()==addB)
{
 result=n1+n2;
}
else if(e.getSource()==subB)
{
 result=n1-n2;
}
else if(e.getSource()==mulB)
{
 result=n1*n2;
}

else if(e.getSource()==divB)
{
 if(n2!=0)
 {
 result=n1/n2;
 }
 else
 {
  result=0;
 }
}
else
{
result=0;
}
resultF.setText(String.valueOf(result));
}
public static void main(String[] args) {
       JFrame frame = new JFrame("Arithmetic Calculator Applet");
       ArithmeticCalculatorApplet applet = new ArithmeticCalculatorApplet();
       applet.init();
       frame.add(applet);
       frame.setSize(300, 200);
       frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       frame.setVisible(true);
   }
}
}
```

```
import tkinter as tk
import random

def changecolor():
    newcolor=f'#{random.randint(0, 0xFFFFFF):06x}'
    root.configure(bg=newcolor)
    root.after(1000,changecolor)

root=tk.Tk()
root.geometry("400x300")
root.title("color chaniging")

changecolor()
root.mainloop()
```

```
class Employee:
    def __init__(self, eid, name, dept, sal):
        self.eid = eid
        self.name = name
        self.dept = dept
        self.sal = sal

    def accept(self):
        self.eid = input("Enter Employee ID: ")
        self.name = input("Enter Employee Name: ")
        self.dept = input("Enter Department: ")
        self.sal = float(input("Enter Salary: "))
    def display(self):
        print(f"Employee ID: {self.eid}")
        print(f"Name: {self.name}")
        print(f"Department: {self.dept}")
        print(f"Salary: {self.sal}")

class Manager(Employee):
    def __init__(self, eid, name, dept, sal, bonus=0):
        super().__init__(eid, name, dept, sal)
        self.bonus = bonus

    def accept(self):
        super().accept()
        self.bonus = float(input("Enter Bonus: "))

    def display(self):
        super().display()
        print(f"Bonus: {self.bonus}")
        print(f"Total Salary (Sal + Bonus): {self.total_salary()}")

    def total_salary(self):
        return self.sal + self.bonus


def find_max_salary_manager(managers):
    max_manager = managers[0]
    for manager in managers:
        if manager.total_salary() > max_manager.total_salary():
            max_manager = manager
    return max_manager

if __name__ == "__main__":
    n = int(input("Enter the number of managers: "))
    managers = []

    # Accept details for n managers
    for _ in range(n):
        print(f"\nEntering details for Manager {_ + 1}:")
        manager = Manager("", "", "", 0)  # Creating an empty Manager object
        manager.accept()  # Accept manager details
        managers.append(manager)  # Add to the list

    # Find the manager with the maximum total salary
    max_salary_manager = find_max_salary_manager(managers)

    # Display the details of the manager with the maximum total salary
    print("\nManager with Maximum Total Salary:")
    max_salary_manager.display()
```