```
2_assi_cpp
//calculator
#include<stdio.h>
#include<iostream>
using namespace std;
struct Calculator{
	//add
	int add(int a,int b){
		return a+b;
	}

	float add(float a,float b){
		return a+b;
	}

	float add(int a,float b){
		return a+b;
	}
	float add(float a,int b){
		return a+b;
	}

	//sub
	int sub(int a,int b){
		return a-b;
	}
	float sub(float a,float b){
		return a-b;
	}
	float sub(int a,float b){
		return a-b;
```

```
}
float sub(float a,int b){
        return a-b;
}

//mul
int mul(int a, int b){
        return a*b;
}

float mul(float a,float b){
        return a*b;
}

float mul(int a,float b){
        return a*b;
}
float mul(float a, int b){
        return a*b;
}
//div
int div(int a,int b){
        return a/b;
}

float div(float a,float b)
{
return a/b;
}

int div(int a,float b){
		return a/b;
	}

	float div(float a,int b){
		return a/b;
	}



};

int main(){

	Calculator c;
	int x=10,y=20;
	float p=20.5,q=5.5;
	cout<<"Add (overloading:)\n\n";
	cout<<"add(int,int)"<<c.add(x,y)<<"\n";
	cout<<"add(float,float):"<<c.add(p,q)<<"\n";
	cout<<"add(int,float):"<<c.add(x,p)<<"\n";
	cout<<"add(float,int):"<<c.add(p,x)<<"\n";

	cout<<"\n\nSub(overloading:)\n\n";



	cout<<"sub(int,int):"<<c.sub(y,x)<<"\n";
	cout<<"sub(float,float):"<<c.sub(p,q)<<"\n";
	cout<<"sub(int,float):"<<c.sub(y,q)<<"\n";
	cout<<"sub(float,int):"<<c.sub(p,x)<<"\n";

	cout<<"\n\nMul (overloading:)\n\n";
```

```
    cout<<"mul(int,int):"<<c.mul(x,y)<<"\n";
    cout<<"mul(flaot,float)"<<c.mul(p,q)<<'\n';
    cout<<"mul(int,float):"<<c.mul(x,q)<<"\n";
    cout<<"mul(flaot,int):"<<c.mul(p,y)<<"\n";

    cout<<"\n\ndiv (overloading:)\n\n"<<"\n";
    cout<<"div(int,int):"<<c.div(y,x)<<"\n";
    cout<<"div(float,float):"<<c.div(p,q)<<"\n";
    cout<<"div(int,float):"<<c.div(y,q)<<"\n";
    cout<<"div(float,int):"<<c.div(p,x)<<"\n";


    return 0;
}
//loan approval
//#include<stdio.h>
#include<string.h>
#include<iostream>
using namespace std;
struct Student{
    int rollNo;
    char name[20];
    double percentage;


    Student(){
        cout<<"Defalut const\n";
        this->rollNo=0;
        strcpy(this->name,"student");
        this->percentage=0;
    }
```

```
Student(int r,char* nm, double p){
        cout<<"Parameterised const\n";
        this->rollNo=r;
        strcpy(this->name,nm);
        this->percentage=p;
}
void setRollno(int r){
        this->rollNo=r;
}

void setName(char* nm){
        strcpy(this->name,nm);
}

void setPercentage(double p){
        this->percentage=p;
}

//getters
int getRollno(){
        return this->rollNo;
}

char* getName(){
        return this->name;
}

double getPercentage(){
        return this->percentage;
}
```

```
		void display(){
			cout<<"Roll No :"<<this->rollNo<<"\n";
			cout<<"Name:"<<this->name<<"\n";
			cout<<"Percentage:"<<this->percentage<<"\n";
		}




	};



		struct Employee{
			int id;
			char name[20];
			double salary;

			Employee(){
				cout<<"Deflaut constructor!\n";
				this->id=0;
				strcpy(this->name,"Employee");
				this->salary=0;

			}

			Employee(int i,char* nm,double s){
			cout<<"Parameterised constructor called\n";
				this->id=i;
				strcpy(this->name,nm);
		this->salary=s;
	}

	void setId(int i){
		this->id=i;
	}

	void setName(char* nm)
	{
		strcpy(this->name,nm);
	}

	void setSalary(double s){
		this->salary=s;
	}

	//getters
	int getId(){
		return this->id;
	}

	char* getname(){
		return this->name;
	}

	double getSalary(){
		return this->salary;
	}

	void display(){
		cout<<"Id:"<<this->id<<"\n";
```

```
            cout<<"Name:"<<this->name<<"\n";
            cout<<"Salary:"<<this->salary<<"\n";
        }


};


struct LoanAprroval{

    double toapproveLoan(Student s){
        double loan;
                if(s.getPercentage()>=80){
                    loan=200000;
                    return loan;
                }
                else{
                        if(s.getPercentage()>=60){
                            loan=100000;
                            return loan;
                        }
                        else{
                                if(s.getPercentage()<=50 &&
s.getPercentage()>=40){
                                    loan=50000;
                                    return loan;
                                }
                                else{
                                    if(s.getPercentage()<40){
                                        loan=0;
                                        return loan;
                                    }
                                }
                        }
                }
                return 0;
    }


    double toapproveLoan(Employee e){
        double loan;
                if(e.getSalary()>=1200000){
                    loan=700000;
                    return loan ;
                }
                else{
                            if(e.getSalary()>=1000000){
                                loan=600000;
                                return loan;
                            }
                            else{
                                    if(e.getSalary()>=600000){
                                        loan=500000;
                                        return loan ;
                                    }
                                    else{

    if(e.getSalary()>=400000){

    loan=400000;
                                            return loan
;
                                        }
                                        else{

    if(e.getSalary()<400000){

    loan=0;

    return loan;
                                            }
                                        }
                                    }
                            }
                    }

                    return 0;
    }

};

int main(){
    Student s1(1,"prachiti",60);
    s1.display();
    Employee e1(101,"hruru",700000);
    e1.display();

    LoanAprroval loanA;
    double loan_S=loanA.toapproveLoan(s1);
    if(loan_S!=0)
    {
        printf("approve loan of :%lf\n",loan_S);
    }
    else{
        printf("Not approve!!");
    }

    double loan_E=loanA.toapproveLoan(e1);
    if(loan_E!=0)
    {
        cout<<"approve loan of :"<<loan_E<<"\n";
    }
    else{
        cout<<"Not approve!!";
    }

    return 0;
}
//area of shapes
#include<stdio.h>
#include<iostream>
using namespace std;
struct Triangle{
    double height;
    double base;

    Triangle(){
        this->height=0;
        this->base=0;
        cout<<"Defalut const\n";
    }

    Triangle(double h,double b){
        this->height=h;
        this->base=b;
        cout<<"Parameterised const\n";
```

```
		}

		void setHeight(double h){
			this->height=h;
		}
		void setBase(double b){
			this->base=b;
		}

		double getHeight(){
			return this->height;
		}

		double getBase(){
			return this->base;
		}

		void display(){
			cout<<"Height:"<<this->height<<"\n";
			cout<<"Base:"<<this->base<<"\n";
		}
	};


	struct Rectangle{
		double length;
		double breadth;

		Rectangle(){
			cout<<"defalut const\n";
			this->length=0;
```

```
        this->breadth=0;
    }

    Rectangle(double l,double b){
        cout<<"Parameterised const\n";
        this->length=l;
        this->breadth=b;
    }

    void setLength(double l){
        this->length=l;
    }
    void setBreadth(double b){
        this->breadth=b;
    }

    double getLength(){
        return this->length;
    }

    double getBreadth(){
        return this->breadth;
    }

    void display(){
        cout<<"Length:"<<this->length<<"\n";
        cout<<"Breadth:"<<this->breadth<<"\n";
    }
};

struct Circle{
        double radius;

        Circle(){
                cout<<"defalut const\n";
                this->radius=0;
        }

        Circle(double r){
                cout<<"Parameterised const\n";
                this->radius=r;
        }

        void setRadius(double r){
                this->radius=r;
        }

        double getRadiud(){
                return this->radius;
        }

        void display(){
                cout<<"Radius:"<<this->radius<<"\n";
        }

};

struct ShapeOperation{

        double calculateArea(Triangle t){
                return (0.5*t.getHeight()*t.getBase());
        }
        double calculateArea(Rectangle r){
                return (r.getLength()*r.getBreadth());
        }

        double calculateArea(Circle c){
                return (3.14*c.radius*c.radius);
        }
};

int main(){

        Triangle t1(10,2);
        t1.display();
        Rectangle r1(5,3);
        r1.display();
        Circle c1(5.3);
        c1.display();

        ShapeOperation S;
        double Area_T=S.calculateArea(t1);
        cout<<"Area of Triangle:"<<Area_T<<"\n";

        double Area_R=S.calculateArea(r1);
        cout<<"Area of Rectangle:"<<Area_R<<"\n";

        double Area_C=S.calculateArea(c1);
        cout<<"Area of Circle:"<<Area_C<<"\n";

        return 0;
}
```