#include <iostream>

#include <math.h>

#include <fstream>

#include <iomanip>

#include <string>

#include <cmath>

#include <sstream>

#include <stdlib.h>

#include <graphics.h>

#include <windows.h>

using namespace std;

class Bat{

private:

double totalLength;

double avgLengthOfBlade;

double avgWidthOfBlade;

const double bladeTohandle=0.67; // Blade to Handle ratio considered constant

double areaOfBlade;

double areaOfHandle;

double handleLength;

double totalArea;

double tensileStrength;

double power;

const double ballMass=5;

double massOfBat;

double acclBat,deflection;

double acclBall;

public:

void set\_handleLength(){

handleLength=totalLength-avgLengthOfBlade;

}

double get\_handleLength(){

return handleLength;

}

void set\_tensileStrength(double t){

tensileStrength=t;

}

void get\_tensileStrength(){

cout<<"Tensile Strength of wood is "<<tensileStrength<<"\n";

}

Bat(double l,double w){

totalLength=l;

avgWidthOfBlade=w;

}// Constructor sets length and width

Bat(double ba,double bta,double bm){

acclBall=ba;

acclBat=bta;

massOfBat=bm;

}// Constructor sets acceleration of bat and ball as well as mass of bat

~Bat(){}

double cal\_Area(){ // calculates the area of the blade avgLengthOfBlade=bladeTohandle\*totalLength;

cout<<"Considering bat to handle ratio as 0.67, length of the blade will be "<<avgLengthOfBlade<<"\n";

areaOfBlade=avgLengthOfBlade\*avgWidthOfBlade;

return areaOfBlade;

}

void cal\_Area(double l,double r){ // calculates area of handle as well as total area

areaOfHandle=2\*3.142\*r\*(r+l);

cout<<"Area of the handle is "<<areaOfHandle<<"\n";

totalArea=areaOfBlade+areaOfHandle;

cout<<"Total area of the bat is "<<totalArea<<"\n";

}

void cal\_Power(){

double sweetSpot[]={totalArea/10,totalArea/15,totalArea/20,totalArea/25};

power=tensileStrength\*sweetSpot[0];

cout<<"Power in the shot when sweetspot is "<<sweetSpot[0]<<"cm^2 = "<<power<<"\n";

power=tensileStrength\*sweetSpot[1];

cout<<"Power in the shot when sweetspot is "<<sweetSpot[1]<<"cm^2 = "<<power<<"\n";

power=tensileStrength\*sweetSpot[2];

cout<<"Power in the shot when sweetspot is "<<sweetSpot[2]<<"cm^2 = "<<power<<"\n";

power=tensileStrength\*sweetSpot[3];

cout<<"Power in the shot when sweetspot is "<<sweetSpot[3]<<"cm^2 = "<<power<<"\n";

}

void batDeflection(){ //// Measures the bat deflection

double ballForce=ballMass\*acclBall;

double batForce=massOfBat\*acclBat;

deflection=(ballForcebatForce)/tensileStreng;

cout<<"The negligible distance by which the bat will be deflected in opposite\ndirection will be "<<deflection<<"\n";

}

bool Aging(double i,double g){

if(i<g)

return true;

return false;

}

string readGivenFile(){ // radius is obtained through the file

string line,radius;

ifstream back("C:/Users/User/Desktop/Spring/EP/Final Project/batWiki.txt");

while (!back.eof()){

getline(back,line);

if(line.find("centimeters")<400){

radius=line.substr(line.find("centimeters")-2,1);

}

}

back.close();

return radius;

}

string readLife(){ // bat life is read through the file

string line,life;

ifstream back("C:/Users/User/Desktop/Spring/EP/Final Project/batWiki.txt");

while (!back.eof()){

getline(back,line);

if(line.find("years")<400){

life=line.substr(line.find("years")-2,1);

}

}

back.close();

return life;

}

};

int main(int argc, char\*\* argv){

initwindow(800,1000);

setcolor(RED);

moveto(400,200);

lineto(360,250); // left bottom corner

moveto(360,250);

lineto(370,253); // right bottom corner

moveto(400,200);

lineto(410,203);

moveto(410,203);

lineto(370,253);

moveto(360,250);

moveto(312,370);

lineto(370,253);

moveto(312,370);

lineto(316,300);

moveto(316,300);

lineto(360,250);

setcolor(RED); // Position of the ball

setfillstyle(SOLID\_FILL, RED);

circle(750, 200, 20);

floodfill(750,200,RED);

delay(2000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(750, 200, 20);

floodfill(750,200,BLACK);

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(700, 210, 20);

floodfill(700,210,RED);

delay(2000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(700, 210, 20);

floodfill(700,210,BLACK);

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(600, 240, 20);

floodfill(600,240,RED);

setcolor(BLACK); // bat Handle and design of man body

moveto(400,200);

lineto(360,250); // left bottom corner

moveto(360,250);

lineto(370,253); // right bottom corner

moveto(400,200);

lineto(410,203);

moveto(410,203);

lineto(370,253);

moveto(360,250);

moveto(312,370);

lineto(370,253);

moveto(312,370);

lineto(316,300);

moveto(316,300);

lineto(360,250);

setcolor(RED);

moveto(410,210);

lineto(350,230);

moveto(350,230);

lineto(353,237);

moveto(410,210);

lineto(413,217);

moveto(413,217);

lineto(353,237);

moveto(353,237);

lineto(250,290);

moveto(250,290);

lineto(295,245);

moveto(295,245);

lineto(350,230);

delay(2000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(600, 240, 20);

floodfill(600,240,BLACK);

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(500, 280, 20);

floodfill(500,280,RED);

setcolor(BLACK);

moveto(410,210);

lineto(350,230);

moveto(350,230);

lineto(353,237);

moveto(410,210);

lineto(413,217);

moveto(413,217);

lineto(353,237);

moveto(353,237);

lineto(250,290);

moveto(250,290);

lineto(295,245);

moveto(295,245);

lineto(350,230);

setcolor(RED);

moveto(400,200);

lineto(360,250); // left bottom corner

moveto(360,250);

lineto(370,253); // right bottom corner

moveto(400,200);

lineto(410,203);

moveto(410,203);

lineto(370,253);

moveto(360,250);

moveto(312,370);

lineto(370,253);

moveto(312,370);

lineto(316,300);

moveto(316,300);

lineto(360,250);

delay(2000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(500, 280, 20);

floodfill(500,280,BLACK);

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(400, 320, 20);

floodfill(400,320,RED);

delay(2000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(400, 320, 20);

floodfill(400,320,BLACK);

setcolor(BLACK);

moveto(400,200);

lineto(360,250); // left bottom corner

moveto(360,250);

lineto(370,253); // right bottom corner

moveto(400,200);

lineto(410,203);

moveto(410,203);

lineto(370,253);

moveto(360,250);

moveto(312,370);

lineto(370,253);

moveto(312,370);

lineto(316,300);

moveto(316,300);

lineto(360,250);

setcolor(RED);

moveto(385,220);

lineto(380,350);

moveto(385,220);

lineto(375,220); //left bottom corner of handle

moveto(375,220);

lineto(360,280);

moveto(360,280);

lineto(380,350);

moveto(375,220);

lineto(375,160);

moveto(375,160);

lineto(385,160);

moveto(385,160);

lineto(385,220);

PlaySound("C:\\Users\\User\\Desktop\\Spring\\EP\\Final Project\\ipl-final-23.wav",NULL,SND\_ASYNC|SND\_FILENAME);

// upto here it is perfect, want to check the position of ball here with time

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(500, 280, 20);

floodfill(500,280,RED);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(500, 280, 20);

floodfill(500,280,BLACK);

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(750, 210, 20);

floodfill(750,210,RED);

delay(1000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(750, 210, 20);

floodfill(750,210,BLACK);

setcolor(RED);

setfillstyle(SOLID\_FILL, RED);

circle(400, 20, 20);

floodfill(400,20,RED);

delay(1000);

setcolor(BLACK);

setfillstyle(SOLID\_FILL, BLACK);

circle(400, 20, 20);

floodfill(400,20,BLACK);

setcolor(GREEN); // Welcome text

settextstyle(2,HORIZ\_DIR,10);

outtextxy(270,10,"W");

delay(500);

outtextxy(308,10,"e");

delay(500);

outtextxy(335,10,"l");

delay(500);

outtextxy(345,10,"c");

delay(500);

outtextxy(365,10,"o");

delay(500);

outtextxy(395,10,"m");

delay(500);

outtextxy(435,10,"e");

delay(500);

setcolor(BLACK);

moveto(385,220);

lineto(380,350);

moveto(385,220);

lineto(375,220); //left bottom corner of handle

moveto(375,220);

lineto(360,280);

moveto(360,280);

lineto(380,350);

moveto(375,220);

lineto(375,160);

moveto(375,160);

lineto(385,160);

moveto(385,160);

lineto(385,220);

delay(1000);

setcolor(GREEN);

setfillstyle(SOLID\_FILL, RED);

circle(400, 180, 20);

circle(394,178,3); // for eyes

circle(406,178,3);

moveto(400,181); // nose

lineto(400,188);

moveto(394,193); // mouth

lineto(406,193);

moveto(400,200);

lineto(400,210);

rectangle(370,210,430,300);

moveto(370,240);

lineto(345,280);

moveto(430,240); // left hand

lineto(455,280);

circle(345,280,5); // bat handle

moveto(340,280);

lineto(310,310); // copied till here

moveto(310,310); //

lineto(313,313); //

moveto(313,313);

lineto(340,283);

moveto(313,313);

lineto(319,319);

moveto(319,319);

lineto(305,305);

moveto(319,319);

lineto(250,370);

moveto(305,305);

lineto(244,353);

moveto(244,353);

lineto(250,370);

for(int i=0;i<=10;i++) // for leg movement in loop

{

setcolor(GREEN);

moveto(385,300);

lineto(365,370);

moveto(415,300);

lineto(415,370);

if(i==0)

delay(4000);

delay(500);

setcolor(BLACK);

moveto(385,300);

lineto(365,370);

moveto(415,300);

lineto(415,370);

setcolor(GREEN);

moveto(385,300);

lineto(385,370);

moveto(415,300);

lineto(435,370);

delay(500);

setcolor(BLACK);

moveto(385,300);

lineto(385,370);

moveto(415,300);

lineto(435,370);

if(i==10){

setcolor(GREEN);

moveto(385,300);

lineto(385,370);

moveto(415,300);

lineto(415,370);

}

}

delay(1000);

setcolor(BLACK);

moveto(340,280);

lineto(310,310);

moveto(310,310);

lineto(313,313);

moveto(313,313);

lineto(340,283);

moveto(313,313);

lineto(319,319);

moveto(319,319);

lineto(305,305);

moveto(319,319);

lineto(250,370);

moveto(305,305);

lineto(244,353);

moveto(244,353);

lineto(250,370);

delay(500);

//////////////// Final Position////////////////

setcolor(GREEN);

moveto(340,280);

lineto(310,250);

moveto(310,250);

lineto(318,245);

moveto(318,245);

lineto(343,280);

moveto(310,250);

lineto(305,255);

moveto(310,250);

lineto(320,240);

moveto(320,240);

lineto(270,160);

moveto(270,160);

lineto(255,170);

moveto(255,170);

lineto(310,250);

while(!kbhit());

closegraph();

double length,width,a,hlength;

double rnumber,period;

double batLife,strength,batAccl,batMass;

const double ballAcc=50;

int i,j;

bool result;

string radius,lifeOfBat;

cout<<"Can you guess the Length of the Bat used by your favourite cricket player? \n";

cin>>length;

cout<<"\n";

cout<<"Well !! Now it's time to set the width of the bat !! \n";

cin>>width;

cout<<"\n";

Bat b1(length,width);

a=b1.cal\_Area();

cout<<"\n";

cout<<"So the area of blade of the bat is nearly "<<a<<" cm^2\n";

b1.set\_handleLength();

hlength=b1.get\_handleLength();

cout<<"\n\n";

cout<<"But the area of blade is not enough, it's necessary to find area of handle too\n";

cout<<"1.Get the Radius of the handle\n2.Obtain the area of handle and total area\n";

cin>>i;

// cout<<"What can be the radius of the handle of the bat?? Difficult to think, but google knows everything !!!\n";

if(i==1){

radius=b1.readGivenFile();

istringstream (radius) >> rnumber;

cout<<"Radius is "<<rnumber<<"\n";

}

cin>>i;

if(i==2){

b1.cal\_Area(hlength,rnumber);

}

cout<<"\n\n";

cout<<"Guess the number of years for which your favourite player is using the same bat !!!\n";

cin>>period;

lifeOfBat=b1.readLife();

istringstream (lifeOfBat) >> batLife;

cout<<"The mentioned file from Wikipedia says that, a bat can be used for "<<batLife<<" years\n";

result=b1.Aging( batLife,period);

if(result==0)

cout<<"There is no problem in using the same bat furthermore\n";

else

cout<<"Your player is using the old bat, just convey him a message to start using new one\n";

cout<<"\n\n\n";

cout<<"Which operation you want to perform?? \n";

cout<<"1.Power in shot\n2.Deflection of the Bat\n\n";

cout<<"Enter either 1 or 2\n";

cin>>j;

if(j==1){

cout<<"\n";

cout<<"What can be the tensile strength of the bat?\n";

cin>>strength;

b1.set\_tensileStrength(strength);

b1.get\_tensileStrength();

b1.cal\_Power();

}

cin>>j;

if(j==2){

cout<<"\n";

cout<<"Ball Acceleration considered is 50 m/s^2\n";

cout<<"Making assumption that bat acceleration is 1/10th of ball acceleration\n";

batAccl=(ballAcc/10);

cout<<"What is the mass of your bat?\n";

cin>>batMass;

Bat b(ballAcc,batAccl,batMass);

b.set\_tensileStrength(400);

b.batDeflection();

}

return 0;

}