Dokument Kelompok C++ Semster 2.

Algoritma & Pemograman

001.// efek fungsi virtual

#include<iostream.h>

#include<conio.h>

#include<string.h>

class mahasiswa

{

private:

long nomor\_induk;

char nama[35];

char jurusan [20];

public:

mahasiswa ();

void inisialisasi (long no\_induk, char \*nama, char \*jurusan);

friend void tampilkan\_data (mahasiswa mhs);

};

void main()

{

mahasiswa mhs;

mhs.inisialisasi(130,"amri ahmad","teknik informatika");

tampilkan\_data(mhs);

getch();

}

mahasiswa::mahasiswa()

{

nomor\_induk=0;

strcpy(nama, "");

strcpy(jurusan, "");

}

void mahasiswa::inisialisasi (long no\_induk, char \*nama, char \*jurusan)

{

mahasiswa::nomor\_induk= no\_induk;

strcpy(mahasiswa::nama, nama);

strcpy(mahasiswa::jurusan, jurusan);

}

void tampilkan\_data (mahasiswa mhs)

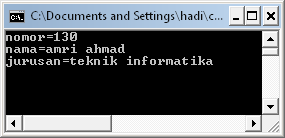
{

cout<<"nomor="<<mhs.nomor\_induk<<endl;

cout<<"nama="<<mhs.nama<<endl;

cout<<"jurusan="<<mhs.jurusan<<endl;

}



002.

#include<iostream.h>

#include<conio.h>

void main()

{

int suhu[8];

cout<<"masukan delapan buah data suhu"<<endl;

for(int i=0; i<8;i++)

{cout<<i+1<<":";

cin>>suhu[i];

}

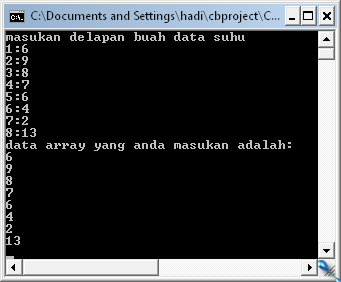
cout<<"data array yang anda masukan adalah:"<<endl;

for(int i=0; i<8;i++)

cout<<suhu[i]<<endl;

getch();

}



003.

#include<iostream.h>

#include<conio.h>

void main(){

char teks[128];

int jumlah=0;

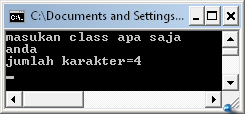
cout<<"masukan class apa saja"<<endl;

cin.get(teks, 128);

cout<<"jumlah karakter="<<cin.gcount()<<endl;

getch();

}



004 #include<iostream.h>

#include<ctype.h>

#include<conio.h>

void main()

{

char slank[15]="bim-bim";

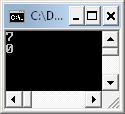
char kosong[15]="";

cout<<strlen(slank)<<endl;

cout<<strlen(kosong)<<endl;

getch();

}.



005. #include<iostream.h>

#include<conio.h>

void main()

{

char nama[125];

cout<<"masukan nama apa saja"<<endl;

cin.get(nama,125);

cout<<" nama mula-mula="<<nama<<endl;

strlwr(nama);

cout<<" nama setelah strlwr="<<nama<<endl;

strupr(nama);

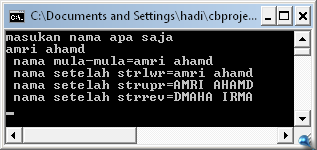
cout<<" nama setelah strupr="<<nama<<endl;

strrev(nama);

cout<<" nama setelah strrev="<<nama<<endl;

getch();

}



006. #include<iostream.h>

#include<conio.h>

void main()

{

int huruf[2][3][3]=

{

{{0,1,0},

{1,1,1},

{1,0,1}

},

{{0,0,1},

{1,0,0},

{0,0,1}

}

};

int i,j,k;

clrscr();

for(i=0;i<2;i++)

{

for (j=0;j<3;j++)

{

for(k=0;k<3;k++)

if(huruf[i][j][k]==1)

cout<<'\*';

else

cout<<'\x25';

cout<<endl;

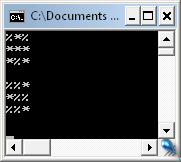
}

cout<<endl;

}

getch();

}



007. #include<iostream.h>

#include<conio.h>

void main()

{

clrscr();//hapus layar

int tgl\_lahir[]={16,8,1987};

int \*ptgl;

ptgl=tgl\_lahir;//ptgl menunjuk ke

//elemen pertama dari array

for (int i=0;i<3;i++)

{

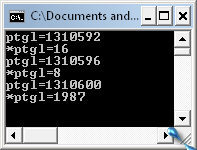
cout<<"ptgl="<<ptgl<<endl;

cout<<"\*ptgl="<<\*ptgl<<endl;

ptgl++; //menunjukan ke elemen berikutnya

}getch();

}



008. #include<iostream.h>

#include<conio.h>

void main()

{

int kode\_bulan;

cout<<"Menentukan bulan"<<endl;

cout<<"kode bulan(1..12):"<<endl;

cout<<"Masukan kode bulan"<<endl;

cin>>kode\_bulan;

switch(kode\_bulan)

{

case 1:

cout<<"Januari"<<endl;

break;

case 2:

cout<<"februari"<<endl;

break;

case 3:

cout<<"maret"<<endl;

break;

case 4:

cout<<"April"<<endl;

break;

case 5:

cout<<"Mei"<<endl;

break;

case 6:

cout<<"Juni"<<endl;

break;

case 7:

cout<<"Juli"<<endl;

break;

case 8:

cout<<"Agustus"<<endl;

break;

case 9:

cout<<"September"<<endl;

break;

case 10:

cout<<"oktober"<<endl;

break;

case 11:

cout<<"November"<<endl;

break;

case 12:

cout<<"Desember"<<endl;

break;

default:

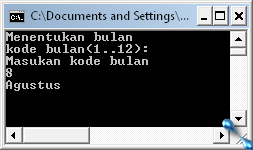
cout<<"Kode hari salah"<<endl;

break;

}

getch();

}



009. #include<iostream.h>

#include<conio.h>

#include<string.h>

class buku

{

public:

char judul[35];

char pengarang [25];

int jumlah;

};

void main()

{

clrscr();

buku novel;

strcpy(novel.judul,"CPP");

strcpy(novel.pengarang,"Amri Ahmad");

novel. jumlah=10;

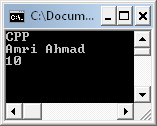
cout<<novel.judul<<endl;

cout<<novel.pengarang<<endl;

cout<<novel.jumlah<<endl;

getch();

}



010. // contoh kelas

// melibatkan anggota data dan fungsi anggota

#include<iostream.h>

#include<conio.h>

#include<string.h>

class buku

{ private:

char judul[35];

char pengarang [25];

int jumlah;

public:

void inisialisasi(char \*Judul, char \*Pengarang, int Jumlah)

{

strcpy(judul,Judul);

strcpy(pengarang,Pengarang);

jumlah=Jumlah;

}

void info()

{

cout<<"Judul="<<judul<<endl;

cout<<"Pengarang="<<pengarang<<endl;

cout<<"Jumlah buku ="<<jumlah<<endl;

}

};

void main()

{

clrscr();

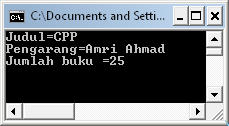
buku novel;

novel.inisialisasi("CPP","Amri Ahmad", 25);

novel. info();

getch();

}



011. // contoh kelas

// memperlihatkan ketik tergantungan anggota pada

// dua buah obyek yang mempunyai kelas yang sama

#include<iostream.h>

#include<conio.h>

#include<string.h>

class buku

{ private:

char judul[35];

char pengarang [25];

int jumlah;

public:

void inisialisasi(char \*Judul, char \*Pengarang, int Jumlah)

{

strcpy(judul,Judul);

strcpy(pengarang,Pengarang);

jumlah=Jumlah;

}

void info()

{

cout<<"Judul="<<judul<<endl;

cout<<"Pengarang="<<pengarang<<endl;

cout<<"Jumlah buku ="<<jumlah<<endl;

}

};

void main()

{

clrscr();

buku novel, fiksi;

novel.inisialisasi("CPP","Amri Ahmad", 25);

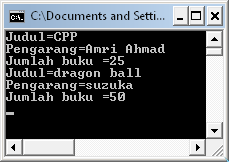
fiksi.inisialisasi("dragon ball","suzuka ", 50);

novel. info();

fiksi. info();

getch();

}



012. // contoh kelas

// penulisan anggota dengan format kelas:: data

#include<iostream.h>

#include<conio.h>

#include<string.h>

class buku

{ private:

char judul[35];

char pengarang [25];

int jumlah;

public:

void inisialisasi(char \*Judul, char \*Pengarang, int Jumlah);

void info();

};

void main()

{

clrscr();

buku novel;

novel.inisialisasi("Kisah seorang pemuda","Budi Hartono", 15);

novel. info();

getch();

}

void buku :: inisialisasi(char \*Judul, char \*Pengarang, int Jumlah)

{

strcpy(buku::judul,Judul);

strcpy(buku::pengarang,Pengarang);

buku::jumlah=Jumlah;

}

void buku :: info()

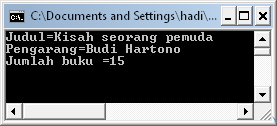
{

cout<<"Judul="<<judul<<endl;

cout<<"Pengarang="<<pengarang<<endl;

cout<<"Jumlah buku ="<<jumlah<<endl;

}



013. // contoh program yang memperlihatkan

// fungsi anggota yang mempunyai nilai balik

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

class Tanggal

{

private :

int tanggal;

int bulan;

int tahun;

public:

void beri\_tanggal(int tanggal, int bulan, int tahun);

void info();

int info\_tanggal() {return tanggal;};

int info\_bulan() {return bulan;};

int info\_tahun() {return tahun;};

};

void main()

{

clrscr();

Tanggal tanggal\_lahir;

tanggal\_lahir. beri\_tanggal(16, 8, 1988);

tanggal\_lahir.info();

cout<<tanggal\_lahir.info\_tanggal()<<'-'

<<tanggal\_lahir.info\_bulan()<<'-'

<<tanggal\_lahir.info\_tahun();

getch();

}

void Tanggal::beri\_tanggal(int tanggal, int bulan, int tahun)

{

Tanggal::tanggal=tanggal;

Tanggal::bulan=bulan;

Tanggal::tahun=tahun;

}

void Tanggal::info()

{

cout<<setfill('0');

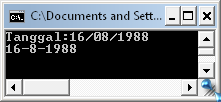
cout<<"Tanggal:"<<setw(2)<<tanggal

<<'/' <<setw(2)<<bulan

<<'/'<<tahun<<endl;

cout<<setfill('0');

}



014. // contoh inisialisasi string

# include<iostream.h>

#include<conio.h>

void main()

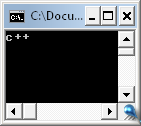
{

char bahasa []="c++";

cout<<bahasa<<endl;

getch();

}



015. //contoh menyalin string dengan strcpy()

#include<iostream.h>

#include<conio.h>

#include<string.h>

void main()

{

char teks[]="C++ Oke";

char data[25];

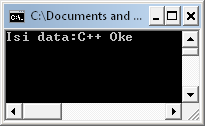
clrscr();

strcpy(data,teks);

cout<<"Isi data:"<<data<<endl;

getch();

}



016. //Contoh pemakaian islower(),isupper(),isdigit(),ispunct(),

#include<iostream.h>

#include<conio.h>

#include<ctype.h>

void main()

{

char st[128];

int jum\_kapital=0,

jum\_kecil=0,

jum\_pungtuasi=0,

jum\_digit=0;

clrscr();

cout<<"Masukan string apa saja"<<endl;

cin.getline(st,sizeof(st));

for(int i=0;st[i];i++)

{

if (islower (st[i]))

jum\_kecil++;

if (isupper (st[i]))

jum\_kapital++;

if (ispunct (st[i]))

jum\_pungtuasi++;

if (isdigit (st[i]))

jum\_digit++;

}

cout<<endl;

cout<<"jumlah"<<endl;

cout<<"huruf kapital:"<<jum\_kapital<<endl;

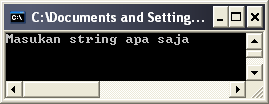
cout<<"huruf kecil:"<<jum\_kecil<<endl;

cout<<"digit:"<<jum\_digit<<endl;

cout<<"pungtuasi"<<jum\_pungtuasi<<endl;

getch();

}



017. //contoh untuk memperlihatkan efek toascii()

#include<iostream.h>

#include<conio.h>

#include<ctype.h>

void main()

{

clrscr();

for(int i=126;i<132;i++)

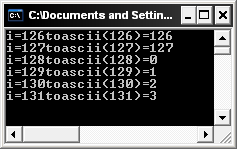
cout<<"i="<<i

<<"toascii("<<i<<")="

<<toascii(i)<<endl;

getch();

}



018. //Mengonversi ke kapital menggunakan toupper()

#include<iostream.h>

#include<conio.h>

#include<ctype.h>

void main()

{

char st[]="penggemar BOLA";

for(int i=0; st[i]; i++)

st[i]=toupper(st[i]);

cout<<st<<endl;

getch();

}



019. // contoh strlen () untuk memperoleh panjang string

# include<iostream.h>

# include<conio.h>

# include<string.h>

void main()

{

char bunga[15]="mawar";

char kosong[15]="";

cout<<strlen(bunga)<<endl;

cout<<strlen(kosong)<<endl;

getch();

}



020. // menunjukan efek strcat

# include<iostream.h>

# include<conio.h>

# include<string.h>

void main()

{ char st[128]="empat,lima";

cout<<"isi st semula:"<<st<<endl;

strcat(st, "enam,tujuh,delapan");

cout<<"isi st sekarang:"<<st<<endl;

getch();

}



021. //efec strstr()

#include<iostream.h>

#include<string.h>

#include<conio.h>

void main()

{

if(strstr("PEMALANG","MALANG"))

cout<<"substring MALANG terdapat pada PEMALANG"<<endl;

if(strstr("PEMALANG","malang"))

cout<<"substring malang terdapat pada PEMALANG"<<endl;

getch();

}



022. // pemakaian strrev()

#include<iostream.h>

#include<string.h>

#include<conio.h>

void main()

{

char kompiler[]="Borlan C++";

strrev(kompiler);

cout<<kompiler<<endl;

getch();

}



023. //mengubah suatu nilai melalui pointer void

#include<iostream.h>

#include<conio.h>

void main()

{

void \*ptr;

int vint=87;

ptr= &vint;

cout<<"vint semula="<<vint<<endl;

\*(int\*)ptr=99;

cout<<"vint sekarang="<<vint<<endl;

getch();

}



024. // pointer menuju ke array

# include<iostream.h>

# include<conio.h>

void main()

{

int tgl\_lahir[]={16, 8, 1988};

int \*ptgl;

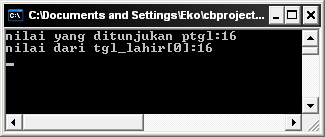
ptgl= tgl\_lahir;

cout<<"nilai yang ditunjukan ptgl:"<<\*ptgl<<endl;

cout<<"nilai dari tgl\_lahir[0]:"<<tgl\_lahir[0]<<endl;

getch();

}



025. // mengakses elemen array via pointer

#include<iostream.h>

#include<conio.h>

void main()

{

int tgl\_lahir[]={16, 8, 1988};

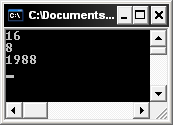
int\*ptgl;

ptgl= tgl\_lahir;

for(int i=0; i<3; i++)

cout<<\*(ptgl+i)<<endl;

getch();

}

026. // mengakses elemen array melalui pointer

#include<iostream.h>

#include<conio.h>

void main()

{

int tgl\_lahir[]={16, 8, 1988};

int\*ptgl;

ptgl= tgl\_lahir;

for(int i=0; i<3; i++)

{

cout<<"ptgl="<<ptgl<<endl;

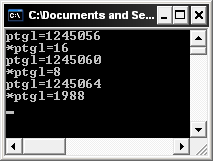
cout<<"\*ptgl="<<\*ptgl<<endl;

ptgl++;

}

getch();

}



027. //contoh array pointer

# include<iostream.h>

# include<conio.h>

void main()

{

char \*nama\_hari[]={"senin","selasa","rabu","kamis","jum'at","sabtu","minggu"};

for(int i=0; i<7; i++)

cout<<nama\_hari[i]<<endl;

getch();

}



028. //pointer yang menuju ke pointer lain

# include<iostream.h>

# include<conio.h>

void main()

{

int var\_x=578;

int \*ptr1;

int \*\*ptr2;

cout<<"var\_x="<<var\_x<<endl;

ptr1=&var\_x;

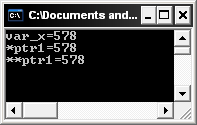
ptr2=&ptr1;

cout<<"\*ptr1="<<\*ptr1<<endl;

cout<<"\*\*ptr1="<<\*\*ptr2<<endl;

getch();

}



029. // argument berupa referensi

#include<iostream.h>

#include<conio.h>

void ubah\_nilai(int &x);

void main()

{

int pinokio= 10;

cout<<"nilai mula- mula untuk pinokio:"<<pinokio<<endl;

ubah\_nilai(pinokio);

cout<<"nilai untuk pinokio sekarang:"<<pinokio<<endl;

getch();

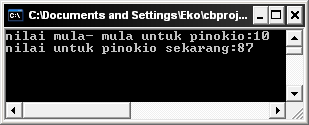
}

void ubah\_nilai(int &x)

{

x=87;

}



030. // fungsi nilai balik berupa pointer

# include<iostream.h>

# include<conio.h>

char \*nama\_bulan(int n);

void main()

{

int bln;

cout<<"bulan(1...12):";

cin>>bln;

cout<<nama\_bulan(bln)<<endl;

getch();

}

char \*nama\_bulan(int n)

{

static char \*bulan[]=

{

"kode bulan salah",

"januari",

"februari",

"maret",

"april",

"mei",

"juni",

"juli",

"agustus",

"september",

"oktober",

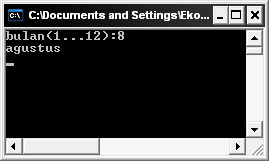
"november",

"desember"

};

return ((n<1||n>12)?bulan[0]:bulan[n]);

}



031. //contoh memberikan gambaran pointer konstant

// mengubah isi yang ditunjukan pointer

// diperkenankan

#include<iostream.h>

#include<conio.h>

void main()

{

char\* const ptr="C++";

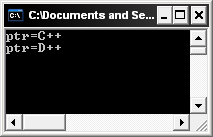
cout<<"ptr="<<ptr<<endl;

\*ptr='D';

cout<<"ptr="<<ptr<<endl;

getch();

}



032. // membaca argument baris perintah

#include<iostream.h>

#include<conio.h>

void main(int argc, char \*argv[])

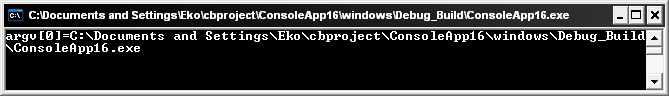
{

for(int i=0; i<argc; i++)

cout<<"argv["<<i<<"]="<<argv[i]<<endl;

getch();

}



033. // contoh untuk memperlihatkan efec konstruktor

#include<iostream.h>

#include<conio.h>

class kompleks

{

private:

double re;

double im;

public:

kompleks();

void info();

};

void main()

{

clrscr();

kompleks a;

a.info();

kompleks b;

b.info();

getch();

}

kompleks::kompleks()

{

cout<<"konstruktor dijalankan..."<<endl;

re=5.2;

im=3.6;

}

void kompleks::info()

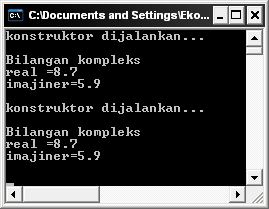
{

cout<<"\nBilangan kompleks"<<endl;

cout<<"real ="<<re<<endl;

cout<<"imajiner="<<im<<endl;

cout<<endl;

}

034. // konstruktor dengan argument

#include<iostream.h>

#include<conio.h>

class buku

{

private:

char judul[35];

char pengarang [25];

int jumlah;

public:

buku(char \*judul, char \*pengarang, int jumlah);

void info();

};

void main()

{

buku fiksi ("si buta dari gua hantu","joko sumarno", 7);

fiksi. info();

getch();

}

buku::buku(char \*judul, char \*pengarang, int jumlah)

{

strcpy(buku::judul, judul);

strcpy(buku::pengarang,pengarang);

buku::jumlah=jumlah;

}

void buku :: info()

{

cout<<"judul="<<judul<<endl;

cout<<"pengarang="<<pengarang<<endl;

cout<<"jumlah="<<jumlah<<endl;

}

035. // memperlihatkan overloading terhadap konstruktor

#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

class berbagimemori

{

private:

union

{

int x;

char y[2];

} mem;

public:

berbagimemori(){mem.x= 0;}

berbagimemori(int A){mem.x=A;}

berbagimemori(char A0 , char A1)

{

mem.y[0]=A0;

mem.y[1]=A1;

}

void tampil();

};

void main()

{

berbagimemori m1;

berbagimemori m2 (0x1234);

berbagimemori m3 (0x27, 0x58);

m1. tampil();

m2. tampil();

m3. tampil();

getch();

}

void berbagimemori::tampil()

{

cout<<setiosflags(ios::hex | ios::showbase);

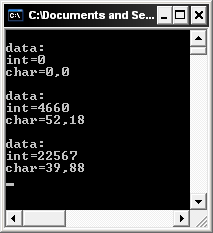
cout<<"\ndata:"<<endl;

cout<<"int="<<mem.x<<endl;

cout<<"char="<<int(mem.y[0])<<","<<int(mem.y[1])<<endl;

cout<<resetiosflags(ios::hex | ios::showbase);

}



036. //Contoh pemberian nilai awal pada konstruktor

#include<iostream.h>

#include<conio.h>

class pencacah

{

private:

int nilai;

public:

pencacah();

void naik();

void tampilkan\_pencacah();

};

void main()

{

pencacah i;

i.tampilkan\_pencacah();

i.naik();

i.tampilkan\_pencacah();

}

pencacah::pencacah():nilai(0)

{

cout<<"inisialisasi pencacah..."<<endl;

}

void pencacah::naik()

{

nilai++;

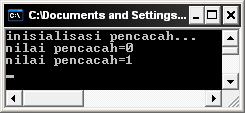
}

void pencacah::tampilkan\_pencacah()

{

cout<<"nilai pencacah="<<nilai<<endl;

getch();

}

037. //pemberian nilai awal terhadap sejumlah anggota data

#include<iostream.h>

#include<conio.h>

class XYZ

{

private:

int x,y,z;

public:

XYZ();

void tampilkan\_xyz();

};

void main()

{

XYZ alpha;

alpha.tampilkan\_xyz();

}

XYZ::XYZ():x(11),y(22),z(33){}

void XYZ::tampilkan\_xyz()

{

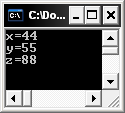
cout<<"x="<<x<<endl;

cout<<"y="<<y<<endl;

cout<<"z="<<z<<endl;

getch();

}



038. //contoh destruktor

#include<iostream.h>

#include<conio.h>

class bilangan

{

private:

int i;

public:

bilangan();

~bilangan();

void info\_data();

};

void main()

{

bilangan x;

x.info\_data();

bilangan y;

y.info\_data();

cout<<"fungsi main()berakhir..."<<endl;

}

bilangan::bilangan()

{

cout<<"konstruktor dijalankan..."<<endl;

i=55;

}

bilangan::~bilangan()

{

cout<<"destruktor dijalankan..."<<endl;

}

void bilangan::info\_data()

{

cout<<"i="<<i<<endl<<endl;

getch();

}

039. //contoh pengalokasian dan pen-dalokasian memori melalui operator new dan delete

#include<iostream.h>

#include<conio.h>

#include<string.h>

class mobil

{

private:

char\*nama;

int cc\_mesin;

int jumlah\_pintu;

public:

mobil(char\*nama\_mobil,int cc,int juml\_pintu);

~mobil();

void keterangan();

};

void main()

{

mobil sedan("charade classy",1300,5);

mobil pickup("datsun",1000,2);

sedan.keterangan();

pickup.keterangan();

}

mobil::mobil(char\*nama\_mobil,int cc,int jum\_pintu)

{

nama=new char[25];

strcpy(nama,nama\_mobil);

cc\_mesin=cc;

jumlah\_pintu = jum\_pintu;

}

mobil::~mobil()

{

delete[]nama;

}

void mobil::keterangan()

{

cout<<"informasi mobil:"<<endl;

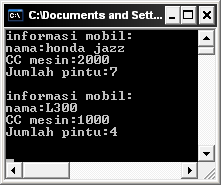
cout<<"nama:"<<nama<<endl;

cout<<"CC mesin:"<<cc\_mesin<<endl;

cout<<"Jumlah pintu:"<<jumlah\_pintu<<endl;

cout<<endl;

getch();



040. //contoh untuk menunjukan obyek lakal dan global

#include<iostream.h>

#include<string.h>

#include<conio.h>

class ABC

{

private:

char nama[20];

public:

ABC(char\*nama);

~ABC();

};

ABC x("x");

void main()

{

cout<<"awal fungsi main()..."<<endl;

ABC y("y");

ABC z("z");

cout<<"akhir fungsi main()..."<<endl;

}

ABC::ABC(char\*nama)

{

strcpy(ABC::nama,nama);

cout<<"konstuktor obyek"<<ABC::nama<<"dijalankan..."<<endl;

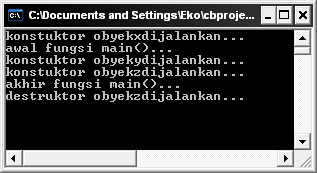
}

ABC::~ABC()

{

cout<<"destruktor obyek"<<nama<<"dijalankan..."<<endl;

getch();

}

041. //contoh untuk memperlihatkan efek exit() pada obuek lokal dan global

#include<iostream.h>

#include<string.h>

#include<stdlib.h>

#include<conio.h>

class ABC

{

private:

char nama[20];

public:

ABC(char\*nama);

~ABC();

};

ABC x("x");

void main()

{

cout<<"awal fungsi main()..."<<endl;

ABC y("y");

ABC z("z");

exit(1);

}

ABC::ABC(char\*nama)

{

strcpy(ABC::nama,nama);

cout<<"konstruktor obyek"<<ABC::nama<<"dijalankan..."<<endl;

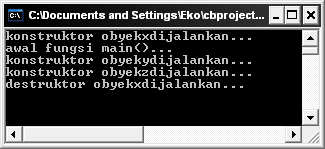
}

ABC::~ABC()

{

cout<<"destruktor obyek"<<nama<<"dijalankan..."<<endl;

getche();

}

042. //fungsi dengan nilai balik objek

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

Buah tambah(Buah b2);

};

void main()

{

Buah buah1(20,5);

Buah buah2(12,4);

Buah buah3;

buah3=buah1.tambah(buah2);

cout<<"obyek buah1"<<endl;

buah1.info\_buah();

cout<<"obyek buah2"<<endl;

buah2.info\_buah();

cout<<"obyek buah3"<<endl;

buah3.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"jumlah apel="<<apel<<"jeruk="<<jeruk<<'\n'<<endl;

}

Buah Buah:: tambah(Buah b2)

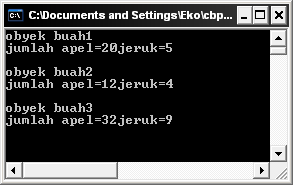
{

Buah tmp;

tmp.apel= apel+ b2.apel;

tmp. jeruk= jeruk + b2.jeruk;

return (tmp);

}

043. //contoh data yang bersifat statis

#include<iostream.h>

#include<conio.h>

class KelasSatatis

{

public:

static int x;

KelasSatatis()

{

x++;

}

void info()

{

cout<<"x="<<x<<endl;

}

};

int KelasSatatis::x=0;

void main()

{

KelasSatatis x,y,z;

x.info();

y.info();

z.info();

KelasSatatis w;

w.info();

cout<<"jumlah obyek berkelas KelasSatatis \n"<<"yang telah diciptakan ="

<< KelasSatatis::x<<"buah"<<endl;

getch();

}

044. // contoh pointer yang menuju ke fungsi anggota

# include<iostream.h>

# include<conio.h>

class KelasX

{

public:

void fung\_1()

{

cout<<"fung\_1() dijalankan..."<<endl;

}

void fung\_2()

{

cout<<"fung\_2() dijalankan..."<<endl;

}

int fung\_3(int a, int b)

{

return(a + b);

}

};

void main()

{

void (KelasX ::\*ptr\_fungsi) (void);

KelasX x;

cout<<"Via fungsi anggota"<<endl;

x.fung\_1();

x.fung\_2();

cout<<"5+5="<<x.fung\_3(5,5)<<endl;

cout<<"\n Via pointer ke fungsi anggota"<<endl;

ptr\_fungsi=&KelasX::fung\_1;

(x.\*ptr\_fungsi) ();

ptr\_fungsi=&KelasX::fung\_2;

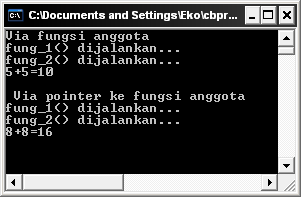
(x.\*ptr\_fungsi) ();

int(KelasX::\*ptr\_fungsi2) (int, int);

ptr\_fungsi2 = &KelasX::fung\_3;

cout<<"8+8="<<(x.\*ptr\_fungsi2)(8,8)<<endl;

getch();

}

045. //overloading terhadap operator +

//dengan bentuk: obyek + konstanta

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

Buah operator + (Buah b2);

Buah operator + (int tambahan);

};

void main()

{

Buah buah1(20,3);

Buah buah2;

cout<<"obyek buah1"<<endl;

buah1.info\_buah();

buah2=buah1+ 3;

cout<<"obyek buah2"<<endl;

buah2.info\_buah();

buah2=buah1+buah1;

cout<<"obyek buah2"<<endl;

buah2.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"jumlah apel="<<apel<<"jeruk="<<jeruk<<'\n'<<endl;

}

Buah Buah:: operator + (Buah b2)

{

Buah tmp;

tmp.apel= apel+ b2.apel;

tmp. jeruk= jeruk + b2.jeruk;

return (tmp);

}

Buah Buah::operator + (int tambahan)

{

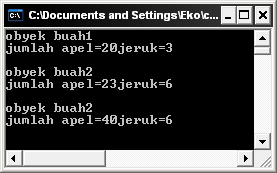
Buah tmp;

tmp.apel= apel+ tambahan;

tmp. jeruk= jeruk + tambahan;

return (tmp);

}



046. //overloading terhadap operator + ,-,\*,/ dan %

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

Buah operator + (Buah b2);

Buah operator - (Buah b2);

Buah operator \* (Buah b2);

Buah operator / (Buah b2);

Buah operator % (Buah b2);

};

void main()

{

Buah buah1(20,8);

Buah buah2 (4,9);

Buah buah3;

cout<<"obyek buah1"<<endl;

buah1.info\_buah();

cout<<"obyek buah2"<<endl;

buah2.info\_buah();

buah3=buah1+buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1+buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1-buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1\*buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1/buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1%buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"jumlah apel="<<apel<<"jeruk="<<jeruk<<'\n'<<endl;

}

Buah Buah:: operator + (Buah b2)

{

Buah tmp;

tmp.apel= apel+ b2.apel;

tmp. jeruk= jeruk + b2.jeruk;

return (tmp);

}

Buah Buah::operator - (Buah b2)

{

Buah tmp;

tmp.apel= apel- b2.apel;

tmp. jeruk= jeruk - b2.jeruk;

return (tmp);

}

Buah Buah::operator \* (Buah b2)

{

Buah tmp;

tmp.apel= apel\* b2.apel;

tmp. jeruk= jeruk \* b2.jeruk;

return (tmp);

}

Buah Buah::operator / (Buah b2)

{

Buah tmp;

tmp.apel= apel/ b2.apel;

tmp. jeruk= jeruk / b2.jeruk;

return (tmp);

}

Buah Buah::operator % (Buah b2)

{

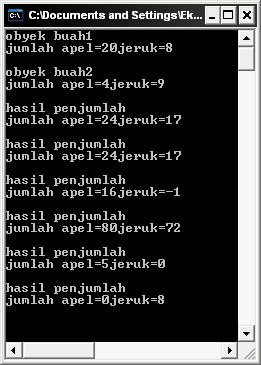
Buah tmp;

tmp.apel= apel% b2.apel;

tmp. jeruk= jeruk % b2.jeruk;

return (tmp);

}



047. //overloading terhadap operator ++ dan --

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

void operator ++ ();

void operator -- ();

};

void main()

{

Buah paket(20,5);

cout<<"Mula- mula"<<endl;

paket.info\_buah();

++paket;

++paket;

cout<<"setelah dinaikan 2x:"<<endl;

paket.info\_buah();

--paket;

cout<<"setelah diturunkan 1x:"<<endl;

paket.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"jumlah apel:"<<apel<<"jeruk:"<<jeruk<<'\n'<<endl;}

void Buah::operator ++()

{

++apel;

--jeruk;

}

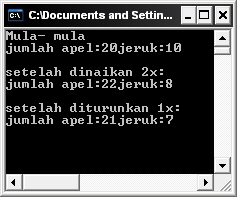
void Buah::operator --()

{

--apel;

--jeruk;

}



048. //overloading terhadap operator ++ versi 2

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

Buah operator ++ ();

};

void main()

{

Buah paket(20,8);

cout<<"obyek paket"<<endl;

paket.info\_buah();

Buah buah = ++paket;

cout<<"setelah : Buah buah = ++paket\n"<<endl;

cout<<"obyek buah"<<endl;

buah.info\_buah();

cout<<"obyek paket"<<endl;

paket.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"Jumlah apel:"<<apel<<"jeruk:"<<jeruk<<'\n'<<endl;}

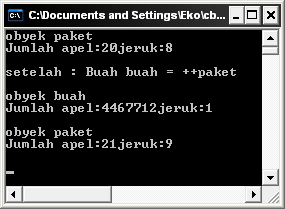
Buah Buah::operator ++()

{

Buah tmp;

tmp. apel = ++apel;

tmp. jeruk = ++jeruk;

}

049. //untuk menggambarkan overloading terhadap operator>

#include<iostream.h>

#include<conio.h>

class Harga

{

private:

long harga\_barang;

float persen\_diskon;

public:

Harga(int=0,float=0.0);

void info\_harga();

int operator<(Harga);

};

void main()

{

Harga barang1(10000,15.0);

Harga barang2(10000,10.0);

Harga barang3(10000,12.5);

if(barang1<barang2)

cout<<"barang1 lebih murah daripada barang2"<<endl;

else

cout<<"barang1 tidak lebih murah daripada barang2"<<endl;

if(barang2<barang3)

cout<<"barang2 lebih murah daripada barang3"<<endl;

else

cout<<"barang2 tidak lebih murah daripada barang3"<<endl;

getch();

}

Harga::Harga(int harga,float diskon)

{

harga\_barang=harga;

persen\_diskon=diskon;

}

void Harga::info\_harga()

{

cout<<"Harga="<<harga\_barang<<endl;

cout<<"Diskon="<<persen\_diskon<<"%\n"<<endl;

}

int Harga::operator<(Harga x)

{

long harga=harga\_barang-

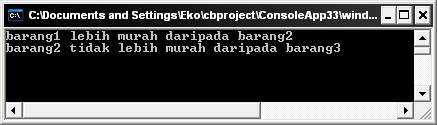
persen\_diskon/100\*harga\_barang;

long harga\_x=x.harga\_barang-

x.persen\_diskon/100\*x.harga\_barang;

return((harga<harga\_x)?1:0);

}



050. //penugasan dengan bentuk:

// obyek\_dengan \_tipe\_data \_dasar

#include<iostream.h>

#include<conio.h>

#include<string.h>

const int MAKS\_KAR=80;

class String

{

private:

char st[ MAKS\_KAR];

public:

String()

{

strcpy(st, "");

}

String(char \*s)

{

strcpy(st, s);

}

void info\_str()

{

cout<<"st="<<st<<endl;

}

};

void main()

{

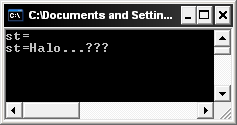
String salam;

salam.info\_str();

salam="Halo...???";

salam.info\_str();

getch();

}

051. //contoh penugasan bentuk

//nilai\_dengan\_tipe\_data\_dasar=obyek

#include<iostream.h>

#include<conio.h>

class jarak

{

private:

int km;

int meter;

public:

jarak (int n\_km, int n\_meter);

operator double()

{

double mil=(1000 \* double (km) +meter)/1600;

return (mil);

}

};

void main()

{

jarak x\_y(115, 7);

double mil =x\_y;

cout<<"jarak dalam satuan mil=" <<mil<<endl;

getch();

}

jarak::jarak(int n\_km, int n\_meter)

{

km= n\_km;

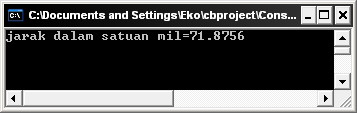
if(n\_meter>1000)

{

km= km + (n\_meter /1000);

meter= (n\_meter % 1000);

}

}

052. //contoh pewarisan kelas berdasarkan program

#include<iostream.h>

#include<conio.h>

#include<string.h>

class kendaraan

{

private:

char nama[15];

public:

kendaraan (char \*nama\_kendaraan ="XXX")

{

strcpy(nama, nama\_kendaraan);

cout<<"hidupkan mesin kendaraan anda..."<<endl;

}

~kendaraan()

{

cout<<"matikan mesin kendaraan anda.."<<endl;

}

void info\_kendaraan()

{

cout<<nama<<"sedang berjalan.."<<endl;

}

};

class truk : public kendaraan

{

public:

truk(char \*nama\_truk) :kendaraan (nama\_truk)

{

cout<<"hidupkan mesin truk.."<<endl;

}

~truk()

{

cout<<"matikan mesin truk anda..."<<endl;

}

};

void main()

{

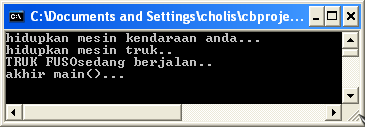
truk fuso ("TRUK FUSO");

fuso.info\_kendaraan();

cout<<"akhir main()..."<<endl;

getch();

}



053. // contoh penggunaan protected

#include <iostream.h>

#include <conio.h>

class basis

{

protected:

int alpha;

int bravo;

public:

void info\_basis()

{

cout<<"info\_basis() dijalankan .. "<<endl;

}

};

class turunan : public basis

{

public:

void inisialisasi (int a, int b)

{alpha = a; bravo = b;}

void info\_turunan()

{

cout<<"alpha="<<alpha<<"bravo="<<bravo<<endl;

}

};

void main()

{

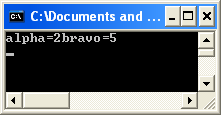
turunan anak;

anak.inisialisasi(2,5);

anak.info\_turunan();

getch();

}



054. // contoh pewrisan kelas berganda

#include<iostream.h>

#include<conio.h>

class bapak

{

public:

bapak()

{

cout<<"konstruktor bapak dijalankan "<<endl;

}

~bapak()

{

cout<<"destruktor bapak dijalankan "<<endl;

}

};

class ibu

{

public:

ibu()

{

cout<<"konstruktor ibu dijalankan "<<endl;

}

~ibu()

{

cout<<"destruktor ibu dijalankan "<<endl;

}

};

class anak

{

public:

anak()

{

cout<<"konstruktor anak dijalankan "<<endl;

}

~anak()

{

cout<<"destruktor anak dijalankan "<<endl;

}

};

void main()

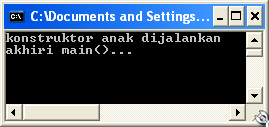
{

anak kandung;

cout<<"akhiri main()..."<<endl;

getch();

}



055. // pengaksesan fungsi anggota yang mengakibatkan keracunan

// pada pewarisan yang beraganda

#include<iostream.h>

#include<conio.h>

class kelasA

{

public:

void info()

{

cout<<"info dari kelas A "<<endl;

}

};

class kelasB

{

public:

void info()

{

cout<<"info dari kelas B" <<endl;

}

};

class kelasC

{

public:

void info()

{

cout<<"info dari kelas C "<<endl;

}

};

void main()

{

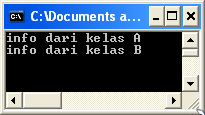
kelasC x;

cout<< "info dari kelas A"<<endl;

cout<< "info dari kelas B"<<endl;

getch();

}



056. #include<iostream.h>

#include<conio.h>

// contoh menunjukan obyek lokal dan global

class ABC

{

private:

char nama[20];

public:

ABC(char \*nama);

~ABC();

};

ABC x("x");

void main()

{

cout<<"awal fungsi main ()..."<<endl;

ABC y("y");

ABC z("z");

cout<<"akhir fungsi main ()..."<<endl;

};

ABC::ABC(char \*nama)

{

strcpy(ABC::nama, nama);

cout<<"konstruktur obyek"<<ABC::nama<<"dijalankan..."<<endl;

}

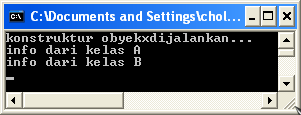
ABC::~ABC()

{

cout<<"destruktor obyek "<<nama<<"dijalankan..."<<endl;

getch();

}



057. //pemakain fungsi yang sama pada kelas

#include<iostream.h>

#include<conio.h>

#include<string.h>

class Orang

{

private:

char nama[50];

int usia;

public:

Orang(char\*nama,int usia);

void info\_orang();

};

class Pegawai:public Orang

{

private:

char bagian[25];

int nomor\_pegawai;

public:

Pegawai(char\*nama,int usia,

char\*bagian,int nomor\_pegawai);

void info\_orang();

};

class Manajer:public Pegawai

{

private:

char mobil[30];

int nomor\_pegawai;

public:

Manajer(char\*nama,int usia,

char\*bagian,int nomor\_pegawai,char\*mobil);

void info\_orang();

};

void main()

{

clrscr();

Manajer kabag\_edp("udin",35,"EDP",11885,"sedan larantuka");

kabag\_edp.info\_orang();

Pegawai staff("siti rukmana",22,"EDP",12345);

staff.info\_orang();

}

Orang::Orang(char\*nama, int usia)

{

strcpy(Orang::nama,nama);

Orang::usia=usia;

}

void Orang::info\_orang()

{

cout<<"Nama:"<<nama<<endl;

cout<<"Usia:"<<usia<<endl;

}

Pegawai::Pegawai(char\*nama,int usia,

char\*bagian,int nomor\_pegawai):

Orang(nama,usia)

{

strcpy(Pegawai::bagian,bagian);

Pegawai::nomor\_pegawai=nomor\_pegawai;

}

void Pegawai::info\_orang()

{

Orang::info\_orang();

cout<<"Bagian:"<<bagian<<endl;

cout<<"Nomor peg.:"<<nomor\_pegawai<<endl;

}

Manajer::Manajer(char\*nama,int usia,

char\*bagian,int nomor\_pegawai,char\*mobil):

Pegawai(nama,usia,bagian,nomor\_pegawai)

{

strcpy(Manajer::mobil,mobil);

}

void Manajer::info\_orang()

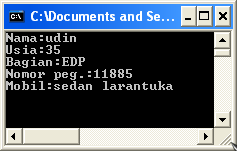
{

Pegawai::info\_orang();

cout<<"Mobil:"<<mobil<<'\n'<<endl;

getch();

}



058. #include <iostream.h>

#define maks 5

class array1D{

friend ostream& operator<<(ostream&,const array1D&);

friend istream& operator>>(istream&,array1D&);

public:

array1D();

void cetak();

void geser\_kiri();

void geser\_kanan();

private:

char A[maks];

};

array1D::array1D(){

for(int i=0;i<maks;i++)

A[i]='0';

}

void array1D::cetak(){

for(int i=0;i<maks;i++)

cout<<A[i]<<" ";

}

ostream& operator<<(ostream& out,const array1D& x){

for(int i=0;i<maks;i++)

cout<<x.A[i]<<" ";

cout<<endl;

return out;

}

istream& operator>>(istream& in,array1D& x){

int posisi;

cout<<"mengisi array pada posisi ke : ";

in>>posisi;

if(posisi>0 && posisi <=maks){

cout<<"masukkan elemen array-nya: ";

in>>x.A[posisi-1];

}

else

cout<<"anda memasukan posisi diluar range...";

return in;

}

void array1D::geser\_kanan(){

int n=maks;

int temp=A[n-1];

for(int i=n-1;i>=0;i--)

A[i+1]=A[i];

A[0]=temp;

}

void array1D::geser\_kiri(){

int n=maks;

int temp=A[0];

for(int i=0;i<n;i++)

A[i]=A[i+1];

A[n-1]=temp;

}

main(){

array1D x;

cout<<"Array masih kosong : "<<x;

cin>>x;

cout<<"isi array saat ini : "<<x;

x.geser\_kiri();

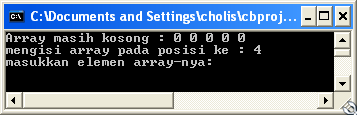
cout<<"Isi array setelah di geser kekiri : "<<x;

x.geser\_kanan();

cout<<"isi array setelah digeser kanan : "<<x;

return 0;

}



059. //Program menentukan zodiac

#include <iostream.h>

#include <conio.h>

class Zodiak{

friend istream& operator>>(istream&, Zodiak&);

friend ostream& operator<<(ostream&, Zodiak&);

public:

Zodiak(){};

int Bintang();

private:

int t;

int b;

};

int Zodiak::Bintang(){

if(b=3&&t>=21,t<=31)

cout<<"Aries";

else if(b=4&&t>=1,t<=19)

cout<<"Aries";

else if(b=4&&t>=20,t<=30)

cout<<"Taurus";

else if(b=5&&t>=1,t<=20)

cout<<"Taurus";

else if(b=5&&t>=21,t<=31)

cout<<"Gemini";

else if(b=6&&t>=1,t<=20)

cout<<"Gemini";

else if(b=6&&t>=21,t<=30)

cout<<"Cancer";

else if(b=7&&t>=1,t<=20)

cout<<"Cancer";

else if(b=7&&t>=23,t<=31)

cout<<"Leo";

else if(b=8&&t>=1,t<=22)

cout<<"Leo";

else if(b=8&&t>=23,t<=31)

cout<<"Virgo";

else if(b=9&&t>=1,t<=22)

cout<<"Virgo";

else if(b=9&&t>=23,t<=30)

cout<<"Libra";

else if(b=10&&t>=1,t<=22)

cout<<"Libra";

else if(b=10&&t>=23,t<=31)

cout<<"Scorpio";

else if(b=11&&t>=1,t<=21)

cout<<"Scorpio";

else if(b=11&&t>=22,t<=30)

cout<<"Sagitarius";

else if(b=12&&t>=1,t<=21)

cout<<"Sagitarius";

else if(b=12&&t>=22,t<=31)

cout<<"Capricorn";

else if(b=1&&t>=1,t<=19)

cout<<"Capricorn";

else if(b=1&&t>=20,t<=31)

cout<<"Aquarius";

else if(b=2&&t>=1,t<=18)

cout<<"Aquarius";

else if(b=2&&t>=19,t<=29)

cout<<"Sagitarius";

else if(b=3&&t>=1,t<=20);

cout<<"Sagitarius";

return 0;

}

istream& operator>>(istream& in,Zodiak& X){

cout<<"Masukkan Tanggal :";

in>>X.t;

cout<<"Masukkan Bulan :";

in>>X.b;

return in;

}

ostream& operator<<(ostream& out,Zodiak& X) {

cout<<"Zodiak anda adalah :";

out<<X.Bintang();

return out;

}

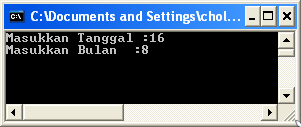
void main(){

Zodiak Bambang ;

cin>>Bambang;

cout<<Bambang;

}



060. //. Program file

#include <iostream.h>

#include <conio.h>

#include <math.h>

class iteratif{

friend istream& operator>>(istream&, iteratif&);

friend ostream& operator<<(ostream&, iteratif&);

public :

iteratif() {};

int fb (int n);

private :

int n;

};

int iteratif::fb(int a){

while ((a==1)||(a==2)){

for (a=1;a<n;a++);

return 1;

}

return(fb(n-1)+fb(n-2));

}

istream& operator>>(istream& in, iteratif& x){

cout<<"Masukkan Nilai n :";

in>>x.n;

return in;

}

ostream& operator<<(ostream& out, iteratif& x){

out<<x.fb( x.n);

return out;

}

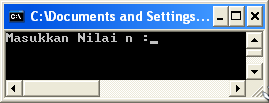
void main(){

iteratif amri;

cin>>amri;

cout<<amri;

}



061. //Program mencetak bilangan 1 sampai dengan 8

#include <iostream.h>

#include <conio.h>

class Cetak{

friend istream& operator<<(istream&,Cetak&);

friend ostream& operator>>(ostream&,Cetak&);

public :

Cetak(){}

int Bilangan();

private :

int bil;

};

int Cetak::Bilangan(){

for(bil=8;bil>=1; bil--)

cout<<bil<<endl;

return 0;

}

istream& operator>>(istream& mlebu,Cetak& M){

cout<<"PROGRAM MENCETAK BILANGAN 1 SAMPAI DENGAN 8"<<endl;

return mlebu;

}

ostream& operator<<(ostream& metu,Cetak& M){

metu<<M.Bilangan();

return metu;

}

void main(){

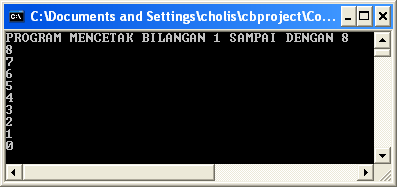
Cetak amri;

cin>>amri;

cout<<amri;

getch();

}



062. //Program volume bangun ruang

#include <iostream.h>

#include <conio.h>

class volume{

friend istream& operator>>(istream&, volume&);

friend ostream& operator<<(ostream&, volume&);

public:

volume(){};

float Kubus ();

float Balok();

float Bola();

private:

int r;

int s;

int p,l,t;

int hasil1;

int hasil2;

float hasil3;

char pil;

};

float volume::Kubus(){

hasil1=s\*s\*s;

cout<<"Masukkan sisi kubus :";cin>>s;

cout<<hasil1;

return hasil1;

}

float volume::Balok(){

hasil2=p\*l\*t;

cout<<"Masukkan panjang :";cin>>p;

cout<<"Masukkan Lebar :";cin>>l;

cout<<"Masukkan tinggi :";cin>>t;

cout<<hasil2;

return hasil2;

}

float volume::Bola(){

hasil3=(4/3.0)\*3.14\*r\*r;

cout<<"Masukkan jari-jari :";cin>>r;

cout<<hasil3;

return hasil3;

}

istream& operator>>(istream& in,volume& x){

cout<<"Program menghitung Volume"<<endl;

cout<<"Masukkan pilihan :";

in>>x.pil;

return in;

}

ostream& operator<<(ostream& out,volume& x){

switch(x.pil){

case'1':x.Kubus();

break;

case'2':x.Balok();

break;

case'3':x.Bola();

}

return out;

}

void main(){

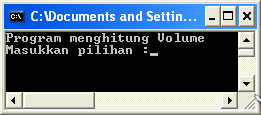
volume hadi;

cin>>hadi;

cout<<hadi;

getch();

}



063. // program menentukan tinggi badab, berat dan umur

// dengan cara menginputkan nya

#include<iostream.h>

#include<conio.h>

void main()

{ int tinggi,umur,berat;

cout<<"Berapakah tinggi badan Anda?";

cin>>tinggi;

cout<<"Tinggi badan Anda adalah"<<tinggi<<".\n";

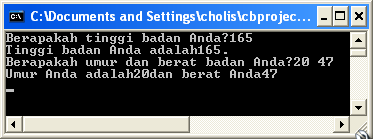
cout<<"Berapakah umur dan berat badan Anda?";

cin>>umur>>berat;

cout<<"Umur Anda adalah"<<umur<<"dan berat Anda"<<berat<<endl;

getche();

}



064. //Program rekursif

#include <iostream.h>

#include <conio.h>

#include <math.h>

class rekursif{

friend ostream& operator<<(ostream&,rekursif&);

friend istream& operator>>(istream&,rekursif&);

public:

rekursif(){};

int fb (int n);

private:

int n;

int x;

};

int rekursif::fb(int n){

if((n==1)||(n==2)) return 1 ;

else return (fb(n-1)+ fb(n-2));

}

istream& operator>>(istream& masuk, rekursif& x){

cout<<"Masukkan Nilai n :";

masuk>>x.n;

return masuk;

}

ostream& operator<<(ostream& keluar, rekursif& x){

cout<<"Hasilnya Adalah :";

keluar<<x.fb(x.n );

return keluar;

}

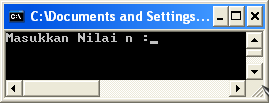
void main(){

rekursif abs;

cin>>abs;

cout<<abs;

getch();

}

065. //pemakain fungsi yang sama pada kelas

#include<iostream.h>

#include<conio.h>

#include<string.h>

class Orang

{

private:

char nama[50];

int usia;

public:

Orang(char\*nama,int usia);

void info\_orang();

};

class Pegawai:public Orang

{

private:

char bagian[25];

int nomor\_pegawai;

public:

Pegawai(char\*nama,int usia,

char\*bagian,int nomor\_pegawai);

void info\_orang();

};

class Manajer:public Pegawai

{

private:

char mobil[30];

int nomor\_pegawai;

public:

Manajer(char\*nama,int usia,

char\*bagian,int nomor\_pegawai,char\*mobil);

void info\_orang();

};

void main()

{

clrscr();

Manajer kabag\_edp("amri",20,"EDP",11885,"suzuki panther");

kabag\_edp.info\_orang();

Pegawai staff("siti rukmana",22,"EDP",12345);

staff.info\_orang();

}

Orang::Orang(char\*nama, int usia)

{

strcpy(Orang::nama,nama);

Orang::usia=usia;

}

void Orang::info\_orang()

{

cout<<"Nama:"<<nama<<endl;

cout<<"Usia:"<<usia<<endl;

}

Pegawai::Pegawai(char\*nama,int usia,

char\*bagian,int nomor\_pegawai):

Orang(nama,usia)

{

strcpy(Pegawai::bagian,bagian);

Pegawai::nomor\_pegawai=nomor\_pegawai;

}

void Pegawai::info\_orang()

{

Orang::info\_orang();

cout<<"Bagian:"<<bagian<<endl;

cout<<"Nomor peg.:"<<nomor\_pegawai<<endl;

}

Manajer::Manajer(char\*nama,int usia,

char\*bagian,int nomor\_pegawai,char\*mobil):

Pegawai(nama,usia,bagian,nomor\_pegawai)

{

strcpy(Manajer::mobil,mobil);

}

void Manajer::info\_orang()

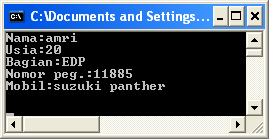
{

Pegawai::info\_orang();

cout<<"Mobil:"<<mobil<<'\n'<<endl;

getch();

}



066. //program menghitung volume bola

#include <iostream.h>

#define phi 3.14

class Bal{

friend istream& operator>>(istream&, Bal&);

friend ostream& operator<<(ostream&, Bal&);

public:

Bal(){};

float Volum();

private:

float r;

float V;

};

float Bal :: Volum(){

V=4/3.0\*phi\*r\*r;

return V;

}

istream& operator>>(istream& Cah, Bal& W){

cout<<"PROGRAM MENGHITUNG VOLUME BOLA "<<endl;

cout<<"Masukkan jari-jari Bola : ";

Cah>>W.r;

return Cah;

}

ostream& operator<<(ostream& Solo, Bal& W){

Solo<<"Volume bola tersebut adalah : "<<W.Volum();

return Solo ;

}

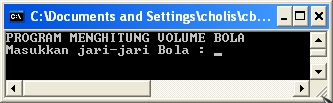
void main(){

Bal amri;

cin>>amri;

cout<<amri;

}



067. //. Program coba nilai

#include <iostream.h>

#include <conio.h>

class Nilai{

friend istream& operator>>(istream&, Nilai&);

friend ostream& operator<<(ostream&, Nilai&);

public:

Nilai(){};

int point();

private:

int N;

char\*Huruf;

};

int Nilai::point(){

if(N<=100 && N>=0){

switch(N/10){

case 10:cout<<"A";break;

case 9:cout<<"A";break;

case 8:cout<<"A";break;

case 7:cout<<"B";break;

case 6:cout<<"C";break;

case 5:cout<<"C";break;

case 4:cout<<"D";break;

case 3:cout<<"D";break;

case 2:cout<<"D";break;

default:cout<<"E";break;

}

}

else

cout<<"Nilai ERROR";

return 0;

}

istream& operator>>(istream& mlebu, Nilai& M){

cout<<"PROGRAM MENGKONVERSI NILAI"<<endl;

cout<<"Masukkan nilai : ";

mlebu>>M.N;

return mlebu;

}

ostream& operator<<(ostream& metu, Nilai& M){

cout<<"Nilai angka yang anda masukkan: ";

metu<<M.N;

cout<<"\nDengan nilai huruf :";

metu<<M.point();

return metu;

}

void main(){

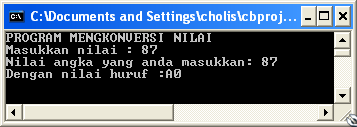
Nilai abs;

cin>>abs;

cout<<abs;

getch();

}



068. //\* melihat pengaruh operator >>

#include<iostream.h>

#include<conio.h>

void main()

{

unsigned int x=87;

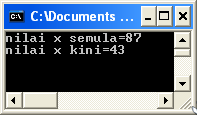
cout<<"nilai x semula="<<x<<endl;

x=x >> 1;

cout<<"nilai x kini="<<x<<endl;

getch();

}



069. //\* melihat pengaruh operator komplement

#include<iostream.h>

#include<conio.h>

void main()

{

unsigned int nilai=81;

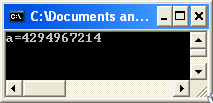
unsigned int a;

a=~nilai;

cout<<"a="<<a<<endl;

getch();

}



070. //\* program mengubah bit 7

//\* dari data bertipe interger agar bernila 1

#include<iostream.h>

#include<conio.h>

void main()

{

unsigned int nilai=87;

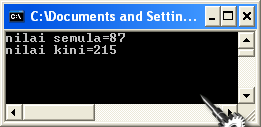
cout<<"nilai semula="<<nilai<<endl;

nilai= nilai| 128;

cout<<"nilai kini="<<nilai<<endl;

getch();

}



071. //\* operasi dengan operator majemuk

#include<iostream.h>

#include<conio.h>

void main()

{

int x=4;

cout<<"x="<<x<<endl;

x+=7;

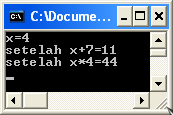
cout<<"setelah x+7="<<x<<endl;

x\*=4;

cout<<"setelah x\*4="<<x<<endl;

getch();

}



072. //\* menampilkan kode ASCII dari suatu karakter

//\* melalui konversi tipe

# include<iostream.h>

# include<conio.h>

void main()

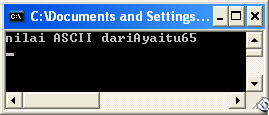
{

char kar ='A';

cout<<"nilai ASCII dari"<<kar<<"yaitu"<<int(kar)<<endl;

getch();

}



073. //\*penggunaan puataka fungsi sqrt()

#include<iostream.h>

#include<conio.h>

#include<math.h>

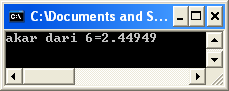
void main()

{

cout<<"akar dari 6="<<sqrt(6)<<endl;

getch();

}



074. //\* mengatur lebar tampilan untuk suatu nilai

#include<iostream.h>

#include<iomanip.h>

#include<conio.h>

void main()

{

int bil= 567;

cout<<setw(0)<<bil<<endl;

cout<<setw(1)<<bil<<endl;

cout<<setw(2)<<bil<<endl;

cout<<setw(3)<<bil<<endl;

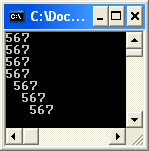
cout<<setw(4)<<bil<<endl;

cout<<setw(5)<<bil<<endl;

cout<<setw(6)<<bil<<endl;

getch();

}



075. //Program menampilkan jumlah hari

#include <iostream.h>

#include <conio.h>

class HARI{

friend istream& operator>>(istream&, HARI&);

friend ostream& operator<<(ostream&, HARI&);

public :

HARI(){};

int bulan();

int hari ();

private :

int B;

int T;

};

int HARI ::bulan(){

if (B==1)

cout<<"januari";

else if (B==2)

cout<<"Februari";

else if (B==3)

cout<<"Maret";

else if (B==4)

cout<<"April";

else if (B==5)

cout<<"Mei";

else if (B==6)

cout<<"Juni";

else if (B==7)

cout<<"Juli";

else if (B==8)

cout<<"Agustus";

else if (B==9)

cout<<"September";

else if (B==10)

cout<<"Oktober";

else if (B==11)

cout<<"November";

else if (B==12)

cout<<"Desember";

else

cout<<"Bulan yang anda masukkan salah ";

return 0;

}

int HARI::hari(){

if (B==8)

cout<<"31 Hari";

else if(B==10)

cout<<"31 Hari";

else if(B==12)

cout<<"31 Hari";

else if(B==2){

if (T%4==0 )

cout<<"29 Hari";

else

cout<<"28 Hari";

}

else if(B%2==0)

cout<<"30 Hari";

else

cout<<"31 Hari";

return 0;

}

istream& operator>>(istream&in, HARI&X){

cout<<"PROGRAM MENAMPILKAN JUMLAH HARI DALAM SUATU BULAN DAN TAHUN"<<endl;

cout<<"Masukkan Bulan :";

in>>X.B;

cout<<"Masukkan Tahun :";

in>>X.T;

return in;

}

ostream& operator <<(ostream& out, HARI&X){

cout<<"Yang anda masukkan adalah :"<<endl;

cout<<"Bulan :";

out<<X.bulan()<<endl;

cout<<"Jumlah hari :";

out<<X.hari()<<endl;

return out;

}

void main(){

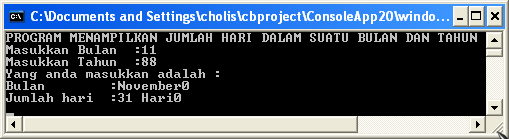
HARI abs;

cin>>abs;

cout<<abs;

getch();

}



076. //Program konversi suhu

#include <iostream.h>

#include<conio.h>

class suhu{

friend istream& operator>>(istream&,suhu&);

friend ostream& operator<<(ostream&, suhu&);

public:

suhu(){};

float R();

float F();

float K();

private:

int C;

float hasil1;

float hasil2;

float hasil3;

};

float suhu::R(){

hasil1=0.8\*C;

return hasil1;

}

float suhu::F(){

hasil2=1.8\*C+32;

return hasil2;

}

float suhu::K(){

hasil3=C+273;

return hasil3;

}

istream& operator>>(istream& in,suhu& x){

cout<<"Program konversi suhu"<<endl;

cout<<"Masukkan suhu derajat celcius : ";

in>>x.C;

return in;

}

ostream& operator<<(ostream& out, suhu& x){

cout<<"suhu dalam Reamur : ";

out<<x.R()<<endl;

cout<<"suhu dalam Fahrenheit : ";

out<<x.F()<<endl;

cout<<"suhu dalam Kelvin : ";

out<<x.K();

return out;

}

void main(){

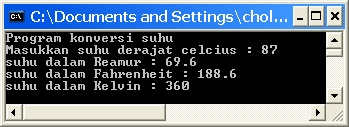
suhu abs;

cin>>abs;

cout<<abs;

getch();

}



077. //Program penghitungan aljabar

#include <iostream.h>

#include <conio.h>

class aljabar{

friend istream& operator>>(istream&,aljabar&);

friend ostream& operator<<(ostream&,aljabar&);

public:

aljabar(){};

float Q();

float R();

private:

int m,n;

float hasil1;

float hasil2;

};

float aljabar::Q(){

hasil1=m/n;

return hasil1;

}

float aljabar::R(){

hasil2=m%n;

return hasil2;

}

istream& operator>>(istream& in,aljabar& x){

cout<<"Program aljabar"<<endl;

cout<<"Masukkan m dan n : "<<endl;

in>>x.m>>x.n;

return in;

}

ostream& operator<<(ostream& out,aljabar& x){

cout<<"nilai q :";

out<<x.Q()<<endl;

cout<<"nilai r :";

out<<x.R();

return out;

}

void main(){

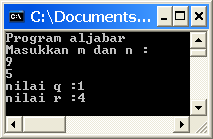
aljabar b;

cin>>b;

cout<<b;

getch();

}



078. //Program menghitung luas segitiga

#include <iostream.h>

#include<conio.h>

class segitiga{

friend istream& operator>>(istream&,segitiga&);

friend ostream& operator<<(ostream&,segitiga&);

public:

segitiga(){};

luas();

private:

int a, t;

int hasil;

};

int segitiga::luas(){

hasil=a\*t/2;

return hasil;

}

istream& operator>>(istream& in,segitiga& x){

cout<<"Program menghitung luas segitiga"<<endl;

cout<<"Masukkan alas dan tinggi : "<<endl;

cout<<"alas : ";

in>>x.a;

cout<<"tinggi :";

in>>x.t;

return in;

}

ostream& operator<<(ostream& out,segitiga& x){

cout<<"luas segitiga tersebut adalah : ";

out<<x.luas();

return out;

}

void main(){

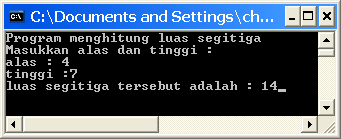
segitiga B;

cin>>B;

cout<<B;

getch();

}



079. //Program menghitung hipotenusa segitiga siku-siku

#include <iostream.h>

#include <math.h>

#include<conio.h>

class Hipotenusa {

friend istream& operator>>(istream&, Hipotenusa&);

friend ostream& operator<<(ostream&, Hipotenusa&);

public:

Hipotenusa(){};

float Sisi();

private:

float a, b, c;

};

float Hipotenusa :: Sisi(){

c=sqrt((a\*a)+(b\*b));

return c;

}

istream& operator>>(istream& mlebu, Hipotenusa& M){

cout<<"PROGRAM MENGHITUNG HIPOTENUSA SEGITIGA SIKU-SIKU"<<endl;

cout<<"Masukkan sisi a : ";

mlebu>>M.a;

cout<<"Masukkan sisi b : ";

mlebu>>M.b;

return mlebu;

}

ostream& operator<<(ostream& metu, Hipotenusa& M){

metu<<"Hipotenusanya adalah : "<<M.Sisi();

return metu;

}

void main(){

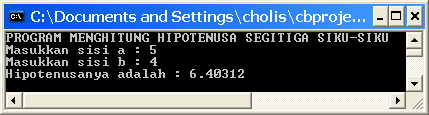
Hipotenusa b;

cin>>b;

cout<<b;

getch();

}



080. // contoh efec manipulaor setprecesion

#include<iostream.h>

#include<iomanip.h>

#include<conio.h>

void main()

{

float nilai=567.89;

clrscr();

cout<<setiosflags(ios::fixed);

cout<<setprecision(0)<<nilai<<endl;

cout<<setprecision(1)<<nilai<<endl;

cout<<setprecision(2)<<nilai<<endl;

cout<<setprecision(3)<<nilai<<endl;

cout<<setprecision(4)<<nilai<<endl;

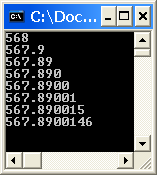
cout<<setprecision(5)<<nilai<<endl;

cout<<setprecision(6)<<nilai<<endl;

cout<<setprecision(7)<<nilai<<endl;

getch();

}



081. //Program mencari nilai maksimum

#include <iostream.h>

#include <conio.h>

class Nilai{

friend istream& operator>>(istream&,Nilai&);

friend ostream& operator<<(ostream&,Nilai&);

public:

Nilai(){};

int maksimum();

private:

int x, y, z;

int hasil;

};

int Nilai::maksimum(){

if(x>y && x>z)

hasil = x;

else if(y>x && y>z)

hasil=y;

else

hasil=z;

return hasil;

}

istream& operator>>(istream& in,Nilai& M){

cout<<"PROGRAM MENENTUKAN NILAI MAKSIMUM "<<endl;

cout<<"Masukkan 3 bilangan bulat (X , Y, Z) : "<<endl;

cout<<"X =";

in>>M.x;

cout<<"Y =";

in>>M.y;

cout<<"Z =";

in>>M.z;

return in;

}

ostream& operator<<(ostream& out,Nilai& M){

cout<<"Nilai maksimum dari ketiga bilangan tersebut adalah : ";

out<<M.maksimum();

return out;

}

void main(){

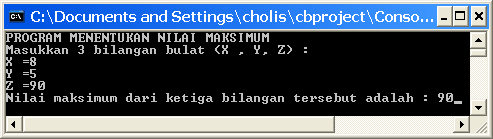
Nilai a;

cin>>a;

cout<<a;

getch();

}



082. // program untuk mengkonverensi

// derajat farenheit ke celcius

// derajat farenhait dimasukan dari keyboard

#include<iostream.h>

#include<conio.h>

void main()

{

int farent, celcius;

cout<<"nilai derajat farentheit: "<<endl;

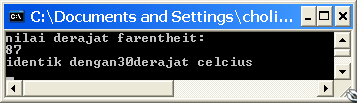
cin>>farent;

celcius=(farent-32)\*5/9;

cout<<"identik dengan"<<celcius<<"derajat celcius"<<endl;

getch();

}



083. //Program selisih waktu

#include <iostream.h>

#include <conio.h>

class Time{

friend istream& operator>>(istream&, Time&);

friend ostream& operator<<(ostream&, Time&);

public:

Time(){};

int waktu1();

int waktu2();

int waktu3();

private:

int j1, j2;

int m1, m2;

int dt1,dt2;

int sel\_detik;

int sel\_menit;

int sel\_jam;

};

int Time :: waktu1(){

if (dt1<dt2)

sel\_detik=dt2-dt1;

else

sel\_detik=(dt2+60)-dt1;

m2=m2-1;

return sel\_detik;

}

int Time :: waktu2(){

if (m2>m1)

sel\_menit=m2-m1;

else

sel\_menit=(m2+60)-m1;

j2=j2-1;

return sel\_menit;

}

int Time :: waktu3(){

if (j2>j1)

sel\_jam=j2-j1;

else

sel\_jam=(j2+24)-j1;

return sel\_jam ;

}

istream& operator>>(istream& mlebu, Time& M){

cout<<"PROGRAM MENGHITUNG SELISIH WAKTU"<<endl;

cout<<"Masukkan waktu awal : "; mlebu>>M.j1>>M.m1>>M.dt1;

cout<<"Masukkan waktu kedua : "; mlebu>>M.j2>>M.m2>>M.dt2;

return mlebu;

}

ostream& operator<<(ostream& metu, Time& M){

cout<<"Selisih waktunya adalah: ";

metu<<M.waktu3()<<"Jam :"<<M.waktu2()<<"Menit :"<<M.waktu1()<<"Detik";

return metu;

}

void main(){

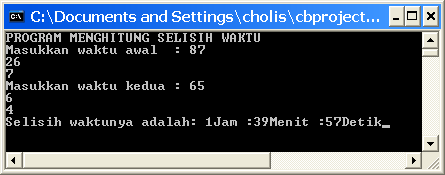
Time a;

cin>>a;

cout<<a;

getch();

}



084. //Program masukan nilai karakter

#include <iostream.h>

#include<conio.h>

class Karakter{

friend istream& operator>>(istream&, Karakter&);

friend ostream& operator<<(ostream&, Karakter&);

public:

Karakter (){};

char huruf();

private:

char H;

};

char Karakter::huruf(){

if(H<='\x5A' && H>='\x41')

cout<<"karakter huruf besar";

else if(H<='\x7A' && H>='\61')

cout<<"karakter huruf kecil";

else if(H<='\x39' && H>='\x30')

cout<<"karakter bilangan";

else

cout<<"Salah Input";

return 0;

}

istream& operator>>(istream& mlebu, Karakter& X){

cout<<"Masukkan karakter : ";

mlebu>>X.H;

return mlebu;

}

ostream& operator<<(ostream& metu, Karakter& X){

metu<<X.huruf();

return metu;

}

void main (){

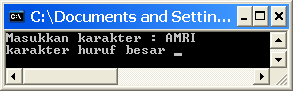
Karakter Amri;

cin>>Amri;

cout<<Amri;

getch();

}



085. //Program konversi detik

#include <iostream.h>

#include<conio.h>

class Konversi{

friend istream& operator>>(istream&, Konversi&);

friend ostream& operator<<(ostream&, Konversi&);

public:

Konversi(){};

int j();

int m();

int d();

private:

int jam;

int menit;

int detik;

int dt;

};

int Konversi :: j(){

jam=dt/3600;

return jam;

}

int Konversi :: m(){

menit=(dt-(3600\*(dt/3600)))/60;

return menit;

}

int Konversi :: d(){

detik=((dt-(3600\*(dt/3600)))-(((dt-(3600\*(dt/3600)))/60)\*(60)));

return detik;

}

istream& operator>>(istream& aku,Konversi& Z){

cout <<"PROGRAM KONVERSI detik MENJADI FORMAT jam:menit:detik"<<endl;

cout<<"masukkan waktu dalam detik: ";

aku>>Z.dt;

return aku;

}

ostream& operator<<(ostream& kamu,Konversi&Z){

kamu<<Z.j()<<"jam : ";

kamu<<Z.m()<<"menit : ";

kamu<<Z.d()<<"detik";

return kamu;

}

void main(){

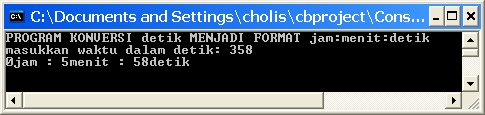
Konversi Amri ;

cin>>Amri;

cout<<Amri;

getch();

}



086. //menentukan bil genap dan ganji dengan pernyataan if

# include<iostream.h>

#include<conio.h>

void main()

{

int bilangan;

cout<<"masukan bilangan bulat"<<endl;

cin>>bilangan;

if(bilangan%2)

{

cout<<"bilangan:"<<bilangan<<"tidak habis dibagi 2"<<endl;

cout<<"jadi termasuk bilangan ganjil"<<endl;

}

else

{

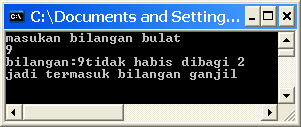
cout<<"bilangan:"<<bilangan<<" habis dibagi 2"<<endl;

cout<<"jadi termasuk bilangan genap"<<endl;

}

getch();

}



087. //\*kode hari.cpp

//\*Contoh pemakaian if untuk menentukan nama hari

#include<iostream.h>

#include<conio.h>

void main()

{

int kode\_hari;

clrscr();

cout<<"Menentukan hari"<<endl;

cout<<"1=SENIN 3=RABU 5=JUM'AT 7=MINGGU"<<endl;

cout<<"2=SELASA 4=KAMIS 6=SABTU"<<endl;

cout<<"kode hari...(1-7):";

cin>>kode\_hari;

if(kode\_hari==1)

cout<<"senin"<<endl;

else if(kode\_hari==2)

cout<<"selasa"<<endl;

else if(kode\_hari==3)

cout<<"rabu"<<endl;

else if(kode\_hari==4)

cout<<"kamis"<<endl;

else if(kode\_hari==5)

cout<<"jum'at"<<endl;

else if(kode\_hari==6)

cout<<"sabtu"<<endl;

else if(kode\_hari==7)

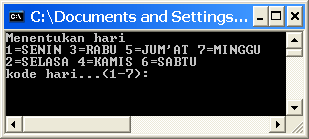
cout<<"minggu"<<endl;

else

cout<<"kode hari salah"<<endl;

getch();

}



088. # include <iostream.h>

#include <conio.h>

#include <iomanip.h>

void garis();

void main()

{

clrscr();

garis();

cout<<setiosflags(ios::left);

cout<<setw(26)<<"judul buku"<<"pengarang"<<endl;

garis();

cout<<setw(26)<<"borland c++"<<"abdul kadir"<<endl;

garis();

}

void garis()

{

int x;

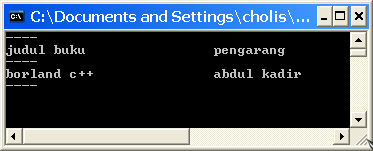
for(x=0;x<4;x++)

cout<<'-';

cout<<endl;

getch();

}



089. #include <iostream.h>

#include <conio.h>

void main()

{

int a=1,b=2,c=3;

cout<<"isi variabel:";

cout<<"a="<<a;

cout<<"b="<<b;

cout<<"c="<<c;

cout<<"\n\nalamat variabel=";

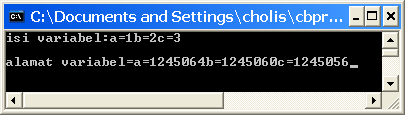
cout<<"a="<<&a;

cout<<"b="<<&b;

cout<<"c="<<&c;

getch();

}



090. // Program mencari titik tengah

#include <iostream.h>

#include<conio.h>

class Titik{

friend istream& operator>>(istream& ,Titik&);

friend ostream& operator<<(ostream& ,Titik&);

public:

Titik(){};

float titik1();

float titik2();

private:

float x1, x2;

float y1, y2;

float TitikTengahx;

float TitikTengahy;

};

float Titik :: titik1(){

TitikTengahx=(x1+x2)/2;

return TitikTengahx;

}

float Titik :: titik2(){

TitikTengahy=(y1+y2)/2;

return TitikTengahy;

}

istream& operator>>(istream& mlebu, Titik& M){

cout<<"PROGRAM MENGHITUNG TITIK TENGAH"<<endl;

cout<<"masukkan nilai x1 dan x2 : ";

mlebu>>M.x1>>M.x2;

cout<<"masukkan nilai y1 dan y2 : ";

mlebu>>M.y1>>M.y2;

return mlebu;

}

ostream& operator<<(ostream& metu, Titik& M){

metu<<"titik tengahnya adalah :("<<M.titik1()<<" , "<<M.titik2()<<")";

return metu;

}

void main(){

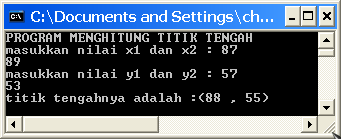
Titik Amri;

cin>>Amri;

cout<<Amri;

getch();

}



091. include <iostream.h>

#include <conio.h>

#include <math.h>

//void main()

//{

// for(int i=1;i<=6;i++)

//cout<<i<<"\t"<<i\*i<<endl;

// getch();

//}

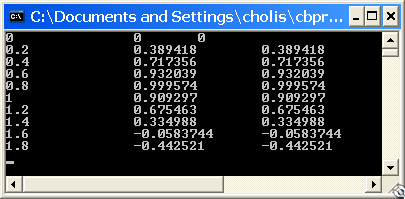
void main()

{for(float x=0; x<2;x+=0.2)

cout<<x<<"\t\t"<<sin(2\*x)<<"\t"<<2\*sin(x)\*cos(x)<<endl;

getch();

}



092. //Program menghitung deret

#include <iostream.h>

#include <conio.h>

class jumlah{

friend istream& operator>>(istream&, jumlah&);

friend ostream& operator<<(ostream&, jumlah&);

public:

jumlah(){};

float deret();

private:

float jml;

int i, n;

};

float jumlah::deret(){

for(i=1; i<=n; i++){

jml==1;

if(i%2==0)

jml=jml-(1/i);

else

jml=jml+(1/i);

}

cout<<jml;

return jml;

}

istream& operator>>(istream& mlebu, jumlah& M){

cout<<"Program menghitung jumlah deret"<<endl;

cout<<"Masukkan nilai n : ";

mlebu>>M.n;

return mlebu;

}

ostream& operator<<(ostream& metu, jumlah& M){

cout<<"\nJumlah deret adalah ";

metu<<M.deret();

return metu;

}

void main(){

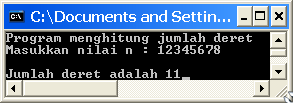
jumlah wahyu;

cin>>wahyu;

cout<<wahyu;

getch();

}



093. #include <iostream.h>

#include <conio.h>

class sigit{

friend ostream& operator<<(ostream&,sigit&);

friend istream& operator>>(istream&,sigit&);

public:

sigit(){};

int fbc (int n);

private:

int n;

int s;

};

int sigit::fbc(int n){

if((n==1)||(n==2)) return 1 ;

else return (fbc(n-1)+ fbc(n-2));

}

istream& operator>>(istream& in, sigit& s){

cout<<"masukkan n =";

in>>s.n;

return in;

}

ostream& operator<<(ostream& out, sigit& s){

cout<<"Hasilnya Adalah :";

out<<s.fbc(s.n );

return out;

}

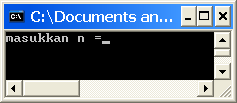
void main(){

sigit Bambang;

cin>>Bambang;

cout<<Bambang;

}



094. //contoh anggota yang bersifat statis

#include<iostream.h>

#include<conio.h>

class KelasStatis

{

public:

static int x;

KelasStatis()

{

x++;

}

void info()

{

cout<<"x="<<x<<endl;

}

};

int KelasStatis::x=0;

void main()

{

clrscr();

KelasStatis x,y,z;

x.info();

y.info();

z.info();

KelasStatis w;

w.info();

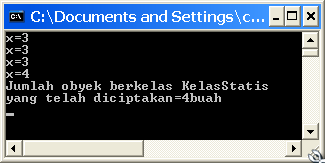
cout<<"Jumlah obyek berkelas KelasStatis\n"

<<"yang telah diciptakan="<<KelasStatis::x

<<"buah"<<endl;

getch();

}



095. #include<iostream.h>

#include<conio.h>

void baca(int a[][5]);

void tulis(const int a[][5]);

int main()

{

int a[3][5];

baca(a);

tulis(a);

return 0;

}

void baca(int a[][5]){

cout<<"masukan 15 angka, 5 bilangan per baris:\n\n";

for(int i=0;i<3;i++){

cout<<"baris"<<i<<":";

for(int j=0;j<5;j++)

cin>>a[i][j];

}

}

void tulis(const int a[][5]){

for(int i=0;i<3;i++){

for(int j=0;j<5;j++)

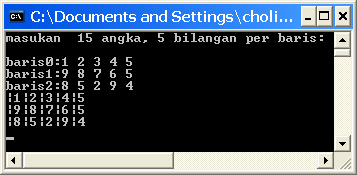
cout<<"|"<<a[i][j];

cout<<endl;

}

getch();

}



096. //overloading terhadap operator ++ versi 2

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

Buah operator ++ ();

};

void main()

{

Buah paket(20,5);

cout<<"obyek paket"<<endl;

paket.info\_buah();

Buah buah = ++paket;

cout<<"setelah : Buah buah = ++paket\n"<<endl;

cout<<"obyek buah"<<endl;

buah.info\_buah();

cout<<"obyek paket"<<endl;

paket.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"Jumlah apel:"<<apel<<"jeruk:"<<jeruk<<'\n'<<endl;}

Buah Buah::operator ++()

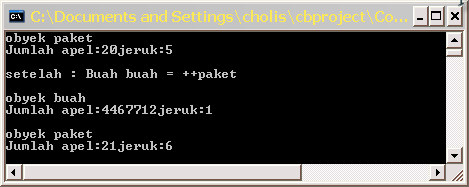
{

Buah tmp;

tmp. apel = ++apel;

tmp. jeruk = ++jeruk;

}



097. //contoh pointer yang menunjuk ke fungsi anngota

#include<iostream.h>

#include<conio.h>

class KelasX

{

public:

void fung\_1()

{

cout << "fung\_1() dijalankan..."<<endl;

}

void fung\_2()

{

cout << "fung\_2() dijalankan..."<<endl;

}

int fung\_3(int a,int b)

{

return(a+b);

}

};

void main()

{

clrscr();

void(KelasX::\*ptr\_fungsi) (void);

KelasX x;

cout<<"Via fungsi anggota"<<endl;

x.fung\_1();

x.fung\_2();

cout<<"5+8="<<x.fung\_3(5,8)<<endl;

cout<<"\nVia pointer ke fungsi anggota"<<endl;

ptr\_fungsi=&KelasX::fung\_1;

(x.\*ptr\_fungsi)();

ptr\_fungsi=&KelasX::fung\_2;

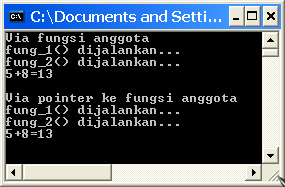
(x.\*ptr\_fungsi)();

int(KelasX::\*ptr\_fungsi2)(int,int);// Ptr ke fungsi

ptr\_fungsi2=&KelasX::fung\_3;

cout<<"5+8="<<(x.\*ptr\_fungsi2)(5,8)<<endl;

getch();



098. //contoh pengonversian dari obyek berkelas

//kartesian ke polar

#include<iostream.h>

#include<conio.h>

#include<math.h>

class Kartesian

{

private:

double x;

double y;

public:

Kartesian(double x=0,double y=0);

void info\_koordinat();

double info\_x();

double info\_y();

};

class Polar

{

private:

double panjang;

double sudut;

public:

Polar(double panjang=0,double sudut=0);

Polar(Kartesian k);

void info\_koordinat();

};

void main()

{

clrscr();

Kartesian posisi1(4,3);

posisi1.info\_koordinat();

Polar posisi2=posisi1;

posisi2.info\_koordinat();

}

Kartesian::Kartesian(double x,double y)

{

Kartesian::x=x;

Kartesian::y=y;

}

void Kartesian::info\_koordinat()

{

cout<<"Koordinat kartesian:"<<endl;

cout<<"x="<<x<<endl;

cout<<"x="<<x<<'\n'<<endl;

}

double Kartesian::info\_x()

{

return(x);

}

double Kartesian::info\_y()

{

return(y);

}

Polar::Polar(double panjang,double sudut)

{

Polar::panjang=panjang;

Polar::sudut=sudut;

}

Polar::Polar(Kartesian k)

{

double x=k.info\_x();

double y=k.info\_y();

panjang=sqrt(x\*x+y\*y);

sudut=atan(y/x);

}

void Polar::info\_koordinat()

{

cout<<"Koordinat polar:"<<endl;

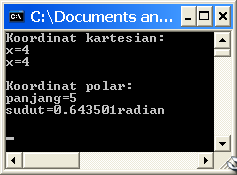
cout<<"panjang="<<panjang<<endl;

cout<<"sudut="<<sudut

<<"radian \n"<<endl;

getch();

}



099. //contoh mengimplementasikan ratio class

#include<iostream.h>

#include<conio.h>

class Pecahan

{

public:

void Isikan(int,int);

double Nilai();

void SeperX();

void Cetak();

private:

int lang,but;

};

int main(){

Pecahan x;

x.Isikan(22,7);

cout<<"x=";

x.Cetak();

cout<<"="<<x.Nilai()<<endl;

x.SeperX();

cout<<"1/x=";x.Cetak();

cout<<endl;

return 0;

}

void Pecahan::Isikan(int pembilang,int penyebut)

{

lang=pembilang;

but=penyebut;

}

double Pecahan::Nilai(){

return double(lang)/but;

}

void Pecahan::SeperX(){

int temp=lang;

lang=but;

but=temp;

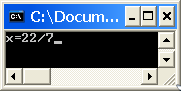
}

void Pecahan::Cetak(){

cout<<lang<<"/"<<but;

getch();

}



100.//overloading terhadap operator + ,-,\*,/ dan %

#include<iostream.h>

#include<conio.h>

class Buah

{

private:

int apel;

int jeruk;

public:

Buah(int jum\_apel=0, int jum\_jeruk=0);

void info\_buah();

Buah operator + (Buah b2);

Buah operator - (Buah b2);

Buah operator \* (Buah b2);

Buah operator / (Buah b2);

Buah operator % (Buah b2);

};

void main()

{

Buah buah1(20,5);

Buah buah2 (3,2);

Buah buah3;

cout<<"obyek buah1"<<endl;

buah1.info\_buah();

cout<<"obyek buah2"<<endl;

buah2.info\_buah();

buah3=buah1+buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1+buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1-buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1\*buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1/buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

buah3=buah1%buah2;

cout<<"hasil penjumlah"<<endl;

buah3.info\_buah();

getch();

}

Buah::Buah(int jum\_apel, int jum\_jeruk)

{

apel= jum\_apel;

jeruk= jum\_jeruk;

}

void Buah::info\_buah()

{

cout<<"jumlah apel="<<apel<<"jeruk="<<jeruk<<'\n'<<endl;

}

Buah Buah:: operator + (Buah b2)

{

Buah tmp;

tmp.apel= apel+ b2.apel;

tmp. jeruk= jeruk + b2.jeruk;

return (tmp);

}

Buah Buah::operator - (Buah b2)

{

Buah tmp;

tmp.apel= apel- b2.apel;

tmp. jeruk= jeruk - b2.jeruk;

return (tmp);

}

Buah Buah::operator \* (Buah b2)

{

Buah tmp;

tmp.apel= apel\* b2.apel;

tmp. jeruk= jeruk \* b2.jeruk;

return (tmp);

}

Buah Buah::operator / (Buah b2)

{

Buah tmp;

tmp.apel= apel/ b2.apel;

tmp. jeruk= jeruk / b2.jeruk;

return (tmp);

}

Buah Buah::operator % (Buah b2)

{

Buah tmp;

tmp.apel= apel% b2.apel;

tmp. jeruk= jeruk % b2.jeruk;

return (tmp);

} 