1. 1. Tuliskan Library dari C++ yang kalian ketahui (serta digunakan untuk apa)!  
   • Beberapa library C++ yang umum digunakan antara lain:  
   • iostream: digunakan untuk input/output stream, seperti cin (input) dan cout (output).  
   • string: digunakan untuk memanipulasi string.  
   • vector: digunakan untuk menyimpan dan mengelola data dalam bentuk array dinamis.  
   • algorithm: berisi berbagai fungsi algoritma, seperti sorting dan searching.  
   • math: berisi fungsi matematika, seperti trigonometri dan eksponensial.

Stdio.h: hamper mirip dengan iostream, untuk input/output, seperti scanf(input) dan printf(output). Biasanya menggunakan pemanggilan tipe data % dan pemanggilan variabel &.  
2. Jelaskan pengertian Public, Protected dan Private (Enkapsulasi)!  
• Public, protected, dan private adalah tiga jenis akses kontrol dalam enkapsulasi di dalam bahasa pemrograman C++. Public memungkinkan sebuah class untuk diakses dari luar, protected memungkinkan class turunan untuk mengakses anggota dari class induk, dan private hanya memungkinkan class itu sendiri untuk mengakses anggota-anggota tersebut. Enkapsulasi memungkinkan class untuk menyembunyikan detail implementasi dari pengguna, sehingga hanya memberikan akses ke fitur-fitur tertentu yang diperbolehkan.  
3. Jelaskan pengertian Inheritance (Pewarisan Class)!  
• Inheritance (pewarisan class) adalah konsep dalam pemrograman berorientasi objek dimana sebuah class dapat mewarisi sifat dan perilaku dari class lain yang sudah ada. Class yang diwarisi disebut sebagai class induk atau parent class, sedangkan class yang mewarisi disebut class turunan atau child class. Dengan inheritance, kita dapat menghindari duplikasi kode dan meningkatkan kemudahan dalam mengelola kode yang kompleks.  
  
4. Jelaskan Pengertian Constructor dan Destructor Object!  
• Constructor dan destructor adalah dua fungsi khusus yang digunakan dalam pemrograman berorientasi objek C++. Constructor digunakan untuk menginisialisasi objek ketika objek tersebut dibuat, sedangkan destructor digunakan untuk membersihkan memori ketika objek dihapus dari memori. Constructor memiliki nama yang sama dengan class, sedangkan destructor memiliki nama yang sama dengan class dengan tambahan simbol tilda (~) di depannya. Constructor dapat memiliki parameter, sedangkan destructor tidak. Constructor akan dipanggil secara otomatis saat objek dibuat, sedangkan destructor akan dipanggil secara otomatis saat objek dihapus dari memori.Tuliskan Code dari Ouput Berikut ini

Text

Description automatically generated

#include<iostream>

#include<stdio.h>

using namespace std;

void Pil(int pil){

int a,b,c;

double d;

cout<<"Masukan Bilangan pertama: ";

scanf("%d",&a);

cout<<"Masukan Bilangan Kedua: ";

scanf("%d",&b);

switch(pil)

{case 1:

c=a+b;

cout<<a<<" + "<<b<<" = "<<c<<endl;

break;

case 2:

c=a - b;

cout<<a<<" + "<<b<<" = "<<c<<endl;

break;

case 3:

c=a\*b;

cout<<c;

break;

case 4:

d = (float)a / (float)b;

cout<<d<<endl;

break;

case 5:

c=a%b;

cout<<c;

break;

default:

cout<<"Masukan dengan benar bung: "<<endl;

break;

}}

int main()

{int pil;

cout<<"1. Penjumlahan\n2. Pengurangan\n3.Perkalian\n4.Pembagian\n5.Modulus\nMasukanPilihan anda: ";

cin>>pil;

Pil(pil);

return 0;

}

1. Lengkapi Code Berikut

Text

Description automatically generated

#include <iostream>

using namespace std;

class Rectangle{

int width, height;

public:

void set\_values(int x, int y);

int area() {

return width \* height;

}

};

void Rectangle::set\_values(int x, int y) {

width = x;

height = y;

}

int main(){

Rectangle rect;

rect.set\_values(12, 4);

cout << "Area: " << rect.area();

return 0;

}

1. Membuat Program kasir sederhana menggunakan ekapsulasi!  
   -banyak jenis barang

-harga barang per jenis

-banyak barang per jenis

-total harga

-bayar

-kembalian

#include <iostream>

#include <string>

using namespace std;

class Barang {

private:

string jenis;

int harga;

int jumlah;

public:

Barang(string jenis, int harga) {

this->jenis = jenis;

this->harga = harga;

this->jumlah = 0;

}

void tambahBarang(int jumlah) {

this->jumlah += jumlah;

}

int getTotalHarga() {

return harga \* jumlah;

}

string getJenis() {

return jenis;

}

};

class Kasir {

private:

Barang \*barang[100];

int jumlahBarang;

int totalHarga;

int bayar;

public:

Kasir() {

jumlahBarang = 0;

totalHarga = 0;

bayar = 0;

}

void tambahBarang(Barang \*barang, int jumlah) {

barang->tambahBarang(jumlah);

totalHarga += barang->getTotalHarga();

this->barang[jumlahBarang++] = barang;

}

int getTotalHarga() {

return totalHarga;

}

int getBayar() {

return bayar;

}

int getKembalian() {

return bayar - totalHarga;

}

void setBayar(int bayar) {

this->bayar = bayar;

}

void cetakStruk() {

cout << "----------------------------------" << endl;

cout << " KFC" << endl;

cout << "----------------------------------" << endl;

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

cout << barang[i]->getJenis() << "\t\t" << barang[i]->getTotalHarga() << endl;

}

cout << "----------------------------------" << endl;

cout << "TOTAL\t\t\t" << getTotalHarga() << endl;

cout << "BAYAR\t\t\t" << getBayar() << endl;

cout << "KEMBALIAN\t\t" << getKembalian() << endl;

cout << "----------------------------------" << endl;

}

};

int main() {

int a, d[100], z[100];

string c[100];

cout << "Masukan Banyak jenis barang: ";

cin >> a;

Barang \*b[a];

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

cout << "Masukan Nama barang ke-" << i + 1 << ": ";

cin >> c[i];

cout << "Masukan Harga barang " << c[i] << ": ";

cin >> d[i];

b[i] = new Barang(c[i], d[i]);

}

Kasir kasir;

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

cout << "Masukan Banyak Barang " << c[i]<< ": ";

cin >> z[i];

kasir.tambahBarang(b[i], z[i]);

}

int o;

cout << "Masukan Uang bayaran: ";

cin >> o;

kasir.setBayar(o);

kasir.cetakStruk();

return 0;

}