|  |  |
| --- | --- |
| Nama | Adidarma Putra Harjuna |
| NRP | 5221600027 |
| Topik | Inheritance |

|  |  |
| --- | --- |
| 1 | 1. Buat kelas Buku yang memiliki field Judul dan Pengarang. Berikan fungsi getter dan  setter dari field tersebut. Berikan pula fungsi untuk menampilkan informasi. Kemudian  buatlah kelas turunan:  a. Fiction: yang memiliki field nilai numerik tingkat membaca. Berikan pula getter  dan setter yang sesuai. Berikan pula fungsi untuk menampilkan informasi.  b. NonFiction, yang memiliki field jumlahHalaman. Berikan pula getter dan setter  yang sesuai. Berikan pula fungsi untuk menampilkan informasi.  Kemudian buat fungsi main() untuk menguji kelas – kelas yang dibuat tersebut. Sipan  dalam file Buku.cpp.  Code :  #include <iostream>  #include <math.h>  #include <string.h>  using namespace std;  //clas Buku  class Buku  {  protected:      string Judul;      string Pengarang;  public:      void setjudul(string);      string getjudul();      void setPengarang(string);      string getPengarang();      void showinfo();  };  void Buku::setjudul(string jdl)  {      Judul = jdl;  }  string Buku::getjudul()  {      return Judul;  }  void Buku::setPengarang(string pgr)  {      Pengarang = pgr;  }  string Buku::getPengarang()  {      return Pengarang;  }  void Buku::showinfo()  {      cout << "Buku Pertama adalah Buku " << Judul << " karya " << Pengarang << endl;  }  //class Fiction  class Fiction : public Buku  {  private:      int nilainumerik;  public:      void setnilainumerik(int);      int getnilainumerik();      void showinfonumerik();  };  void Fiction::setnilainumerik(int numerik)  {      nilainumerik = numerik;  }  int Fiction::getnilainumerik()  {      return nilainumerik;  }  void Fiction::showinfonumerik()  {      cout << "Nilai Numerik Membaca : " << getnilainumerik() << endl;  }  //Class NonFiction  class NonFiction: public Buku  {  private:      int jumlahhalaman;  public:      void showHalaman();      void sethalaman(int);      int gethalaman();  };  void NonFiction::sethalaman(int halaman)  {      jumlahhalaman = halaman;  }  int NonFiction:: gethalaman()  {      return jumlahhalaman;  }  void NonFiction::showHalaman()  {      cout << "Halaman : " << gethalaman() << endl;  }  main()  {      Fiction aBuku;      aBuku.setjudul("Kisah Raden Wijaya");      aBuku.setPengarang("Adi");      aBuku.setnilainumerik(80);      aBuku.showinfo();      aBuku.showinfonumerik();      NonFiction bBuku;      bBuku.setjudul("Kisah Soeharto");      bBuku.setPengarang("M. Andhika");      bBuku.sethalaman(268);      bBuku.showinfo();      bBuku.showHalaman();  };  Output :    Kesimpulan:  Turunan atau child dapat mewarisi beberapa attribute dari base class |
| 2 | 2. Gunakan kelas Car dan RaceCar yang dibuat pada modul You Do It Inheritance dan  modifikasi sesuai petunjuk berikut :  a. Tambahkan field pada kelas Car untuk menyimpan nomor identifikasi  kendaraan dan jumlah miles(jarak) dari kendaraan.  b. Tambahkan fungsi setter untuk nomor id kendaraan. Tambahkan fungsi untuk  menaikkan jumlah jarak(miles). Fungsi ini menambahkan kecepatan  kendaraan sekarang (dalam miles/hour) ke jumlah miles. Kemudian  tambahkan fungsi yang mengembalikan jarak kendaraan (fungsi getter jarak).  c. Modifikasi fungsi turnIgnitionOn() untuk menginisialisasi jarak tempuh ke 0.  Modifikasi fungsi showCar() untuk menambahkan info nomor id dan jarak  tempuh (miles) kendaraan.  d. Tulis fungsi main() yang mendeklarasikan Car dan RaceCar. Set jarak konstan  lintasan balapan misal 500 miles. Kemudian :  i. Berikan no id ke kendaraan,  ii. start kendaraan dan karena kendaraan tidak dapat melebihi jarak maks  lintasan,  iii. buatlah angka acak dibawah 200 untuk mengeset kecepatan dua mobil  tersebut, dan menaikkan jarak tempuh kendaraan (lakukan Langkah iii  berulang).  iv. Ketika salah satu kendaraan melebihi jarak lintasan maka kendaraan  tersebut menang.  Simpan dalam file Race.cpp  Hasil:  #include <iostream>  #include <stdlib.h>  #include <ctime>  using namespace std;  class Car  {  protected:      bool isIgnitionOn;      int speed;      int noplat;      int jarak;  public:      void turnIgnitionOn();      void turnIgnitionOff();      void setSpeed(int);      void showCar();      int getplat();      void setplat(int);      int getjarak();      void setjarak(int);  };  // a,b  int Car::getplat()  {      return noplat;  }  void Car::setplat(int nomor)  {      noplat = nomor;  }  int Car::getjarak()  {      return jarak;  }  void Car::setjarak(const int distance)  {      if (isIgnitionOn)      {          if (jarak <= distance)          {              jarak += speed;          }          else          {              jarak = distance;          }      }      else      {          cout<<"kendaraan tak nyala"<<endl;      }  }  void Car::showCar()  {      // c      cout << "nomor identifikasi = " << getplat() << endl;      cout << "jarak = " << getjarak() << endl;      if (isIgnitionOn)          cout << "Ignition is on. ";      else          cout << "Ignition is off. ";      cout << "Speed is " << speed << endl;  }  void Car::turnIgnitionOn()  {      isIgnitionOn = true;      jarak = 0;  }  void Car::turnIgnitionOff()  {      speed = 0;      isIgnitionOn = false;  }  void Car::setSpeed(int mph)  {      const int STD\_LIMIT = 200;      if (isIgnitionOn)          if (mph <= STD\_LIMIT)              speed = mph;          else              speed = STD\_LIMIT;      else          cout << "Can't set speed - ignition is off!" << endl;  }  class RaceCar : public Car  {  public:      void setSpeed(int mph);  };  void RaceCar::setSpeed(int mph)  {      const int MAX\_SPEED = 200;      if (isIgnitionOn)          if (mph <= MAX\_SPEED)              speed = mph;          else              speed = MAX\_SPEED;      else          cout << "Can't set speed - ignition is off!" << endl;  }  int main()  {      srand((unsigned) time(NULL));      Car aCar;      RaceCar aRaceCar;      int Carspeed = 210;      const int Car\_distance = 500;        aCar.setplat(2068);      aCar.turnIgnitionOn();      Carspeed=rand()%200;      aCar.setSpeed(Carspeed);      aCar.setjarak(Car\_distance);      aCar.showCar();        aRaceCar.setplat(2068);      aRaceCar.turnIgnitionOn();      Carspeed=rand()%200;      aRaceCar.setSpeed(Carspeed);      aRaceCar.setjarak(Car\_distance);      aRaceCar.showCar();      cout<<"\n"<<endl;      if(aCar.getjarak()>aRaceCar.getjarak())      {          cout<<"a car menang"<<endl;      }      else{          cout<<"race car menang"<<endl;      }       return 0;  }  Output : |