Informatik Labore Code:

Labor 1.1 Hello World:

#include <stdio.h>  
  
int main() {  
 printf("Hello, World!\n");  
 return 0;  
}

Labor 1.2 Zahlensysteme mit Printf und Scanf:

#include <stdio.h>  
int \* integer;  
int pointer\_to\_integer = &integer;  
//Functions  
/\*int ASCII(int foo) {  
 return ???  
}\*/  
int main() {  
 //pointer\_to\_integer = 10;  
 printf("Please enter a number:\n");  
 scanf("%d", &pointer\_to\_integer);  
 printf("Dez, %d, ASCII %c, Oct: %o, Hex: %x !\n", pointer\_to\_integer, pointer\_to\_integer, pointer\_to\_integer,pointer\_to\_integer);  
 return 0;  
}

Labor 1.3 Taschenrechner:

#include <stdio.h>  
double Number1 = 33234432.000000; //Gives Number1 an arbritrary value for error detection  
char Operation = '?'; //Gives Operation an arbritrary value for error detection  
double Number2 = 33234432.000000; //Gives Number2 an arbritrary value for error detection  
char Plus = '+';  
char Minus = '-';  
char Times = '\*';  
char Div = '/';  
  
int main() {\  
 printf("Enter number, operation and the second number\n");  
 scanf("%lf %c %lf", &Number1, &Operation, &Number2);  
 if ((Operation=='?') || ((Number1==33234432) || (Number2==33234432))){ //Error detection works because scanf will not scan unreadable vals  
 printf("Invalid Operation\n");  
 } else if ((Number1==33234432) || (Number2==33234432)){ //so Number1, Number2 and Operation should keep arbritrary vals  
 printf("Invalid Syntax\n");  
 } else {  
 if (Operation == Plus) {  
 printf("%g", (Number1 + Number2));  
 }  
 if (Operation == Minus) {  
 printf("%g", (Number1 - Number2));  
 }  
 if (Operation == Times) {  
 printf("%g", (Number1 \* Number2));  
 }  
 if (Operation == Div) {  
 if (Number2 == 0){  
 printf("Forbidden: div by 0");  
 }else {  
 printf("%g", (Number1 / Number2));  
 }  
 }  
 }  
 return 0;  
}

Labor 1.4 Kegel/Zylinder Volumen und Area

#include <stdio.h>  
#include <math.h>  
  
int volora;  
int radius;  
int cylinderHeight;  
int coneHeight;  
double pi = 3.14;  
double cylinderVol(double rad,double heightt);  
double coneVol(double rad, double height);  
double cylinderArea(double rad,double heightt);  
double coneArea(double rad, double height);  
int main() {  
 printf("Enter 1 for Area \nEnter 2 for Volume \n");  
 scanf("%i", &volora);  
 printf("Hello, please enter radius\n");  
 scanf("%i", &radius);  
 printf("Please enter cylinder height\n");  
 scanf("%i", &cylinderHeight);  
 printf("Please enter cone height\n");  
 scanf("%i", &coneHeight);  
 printf("calculating...");  
 if (volora == 2) {  
 printf("\n \*\*\* \n \*\*\* \n \*\*\* \n \*\*\* \nThe volume is %f", (cylinderVol(radius, cylinderHeight) + coneVol(radius, coneHeight)));  
 return 0;  
 } else {  
 printf("Area is: %f", (coneArea(radius, coneHeight)+cylinderArea(radius, cylinderHeight)));  
 }  
}  
  
double coneVol(double rad, double height) {  
 return ((pi\*rad\*rad\*height)/3);  
}  
double cylinderVol(double rad, double heightt) {  
 return((pi\*rad\*rad\*heightt));  
}  
double coneArea(double rad, double height) {  
 return (pi\*rad\*(rad+sqrt(height\*height+radius\*radius)));  
}  
double cylinderArea(double rad, double heightt) {  
 return((2\*pi\*rad\*heightt)+2\*pi\*rad\*rad);  
}

Labor 1.5 Automobile Struct

#include <stdio.h>  
  
int main () {  
 typedef struct {  
 char \*make;  
 char \*model;  
 int year;  
 int disp;  
 int power;  
 int topSpeed;  
 } Automobile\_t;  
 //Preset Input Car1  
 Automobile\_t predefinedAutomobileOne;  
 predefinedAutomobileOne.make = "Ford";  
 predefinedAutomobileOne.model = "Taurus";  
 predefinedAutomobileOne.year = 2002;  
 predefinedAutomobileOne.disp = 2200;  
 predefinedAutomobileOne.power = 155;  
 predefinedAutomobileOne.topSpeed = 190;  
  
 //Preset Input Car2  
 Automobile\_t secondAutomobile;  
 secondAutomobile.make = "Ford";  
 secondAutomobile.model = "Mustang";  
 secondAutomobile.year = 2018;  
 secondAutomobile.disp = 4998;  
 secondAutomobile.power = 400;  
 secondAutomobile.topSpeed = 270;  
  
 printf("%s\n%s\n%d\n%d\n%d\n%d\n", predefinedAutomobileOne.make, predefinedAutomobileOne.model, predefinedAutomobileOne.year, predefinedAutomobileOne.disp, predefinedAutomobileOne.power, predefinedAutomobileOne.topSpeed);  
 printf("%s\n%s\n%d\n%d\n%d\n%d\n", secondAutomobile.make, secondAutomobile.model, secondAutomobile.year, secondAutomobile.disp, secondAutomobile.power, secondAutomobile.topSpeed);  
 return 0;  
}

Labor 1.5 Erweiterte Version (mit nutzereingabe):

#include <stdio.h>  
  
int main () {  
typedef struct {  
 char \*make;  
 char \*model;  
 int year;  
 int disp;  
 int power;  
 int topSpeed;  
} Automobile\_t;  
char buck[10];  
char brick[10];  
 //Preset Input Car1  
 /\*Automobile\_t predefinedAutomobileOne;  
 predefinedAutomobileOne.make = "Ford";  
 predefinedAutomobileOne.model = "Taurus";  
 predefinedAutomobileOne.year = 2002;  
 predefinedAutomobileOne.disp = 2200;  
 predefinedAutomobileOne.power = 155;  
 predefinedAutomobileOne.topSpeed = 190;\*/  
  
 //Preset Input Car2  
 /\*Automobile\_t secondAutomobile;  
 secondAutomobile.make = "Ford";  
 secondAutomobile.model = "Mustang";  
 secondAutomobile.year = 2018;  
 secondAutomobile.disp = 4998;  
 secondAutomobile.power = 400;  
 secondAutomobile.topSpeed = 270;\*/  
  
 //User Input Automobile  
 Automobile\_t firstAutomobile;  
 printf("Please enter make\n");  
 gets(buck);  
 firstAutomobile.make = buck;  
 printf("Please enter model\n");  
 gets(brick);  
 firstAutomobile.model = brick;  
 printf("Please enter year\n");  
 scanf("%d", &firstAutomobile.year);  
 printf("Please enter displacement\n");  
 scanf("%d", &firstAutomobile.disp);  
 printf("Please enter power\n");  
 scanf("%d", &firstAutomobile.power);  
 printf("Please enter top speed\n");  
 scanf("%d", &firstAutomobile.topSpeed);  
 printf("Loading...\n");  
 printf("%s\n%s\n%d\n%d\n%d\n%d\n", firstAutomobile.make, firstAutomobile.model, firstAutomobile.year, firstAutomobile.disp, firstAutomobile.power, firstAutomobile.topSpeed);  
 return 0;  
}

Labor 1.6 Enumerationstypen und Switch Case:

#include <stdio.h>  
  
enum colors{*black*, *silver*, *red*, *orange*, *brown*, *blue*, *green*}; //black=0,silver=1...  
  
const char\* getColorName(enum colors color)  
{  
switch (color){  
case (*black*): return "Black";  
case (*silver*): return "Silver";  
case (*red*): return "Red";  
case (*orange*): return "Orange";  
case (*brown*): return "Brown";  
case (*blue*): return "Blue";  
case (*green*): return "Green";  
}  
}  
  
int main () {  
 typedef struct {  
 char \*make;  
 char \*model;  
 int year;  
 int disp;  
 int power;  
 int topSpeed;  
 enum colors color;  
 } Automobile\_t;  
 //Preset Input Car1  
 Automobile\_t predefinedAutomobileOne;  
 predefinedAutomobileOne.make = "Ford";  
 predefinedAutomobileOne.model = "Taurus";  
 predefinedAutomobileOne.year = 2002;  
 predefinedAutomobileOne.disp = 2200;  
 predefinedAutomobileOne.power = 155;  
 predefinedAutomobileOne.topSpeed = 190;  
 predefinedAutomobileOne.color = 1;  
 //Preset Input Car2  
 Automobile\_t secondAutomobile;  
 secondAutomobile.make = "Ford";  
 secondAutomobile.model = "Mustang";  
 secondAutomobile.year = 2018;  
 secondAutomobile.disp = 4998;  
 secondAutomobile.power = 400;  
 secondAutomobile.topSpeed = 270;  
 secondAutomobile.color = 2;  
  
 //Function that translates the enum numbers to strings (color names):  
 printf("%s\n%s\n%d\n%d\n%d\n%d\n%s\n", predefinedAutomobileOne.make, predefinedAutomobileOne.model,  
 predefinedAutomobileOne.year, predefinedAutomobileOne.disp, predefinedAutomobileOne.power,  
 predefinedAutomobileOne.topSpeed, getColorName(predefinedAutomobileOne.color));  
 printf("\n%s\n%s\n%d\n%d\n%d\n%d\n%s\n", secondAutomobile.make, secondAutomobile.model,  
 secondAutomobile.year, secondAutomobile.disp, secondAutomobile.power, secondAutomobile.topSpeed,  
 getColorName(secondAutomobile.color));  
 return 0;  
}