**Ivan Chukharev**

**MAC125.5753**

**03/17/2023**

**In-Class Exercises**

**Topics: Structures**

**Due: 03/24/2023**

**In-Class Exercises: Structures**

|  |  |  |
| --- | --- | --- |
| 1/3a | | #include <iostream>  using namespace std;  struct Date  {  int month;  int day;  int year;  };  struct Time  {  int hours;  int minutes;  int seconds;  };  void print\_date(Date d) {  cout << "Curent date is " << d.month << "/" << d.day << "/" << d.year << endl;  }  void print\_time(Time t) {  cout << "Curent time is " << t.hours << ":" << t.minutes << ":" << t.seconds << endl;  }  int main()  {  Date date;  Time time;  cout << "Input the current month:";  cin >> date.month;  cout << "Input the current day:";  cin >> date.day;  cout << "Input the current year:";  cin >> date.year;      cout << "Input the hour:";  cin >> time.hours;  cout << "Input the minutes:";  cin >> time.minutes;  cout << "Input the seconds:";  cin >> time.seconds;  cout << endl;  print\_date(date);  print\_time(time);  system("Pause");  return 0;  }  **Output:** |
| 2 | | #include <iostream>  using namespace std;  struct Stocks  {  string name;  double share;  int ratio;  };  const int number\_stocks = 5;  void print\_stocks(Stocks s[]) {  for(int i = 0; i < number\_stocks; i++)  cout << s[i].name << " stock is (" << s[i].share << ") x (" << s[i].ratio << ") = " << (s[i].share \* s[i].ratio) << endl;  }  int main()  {  Stocks stock[number\_stocks];    for (int i = 0; i < number\_stocks; i++) {  cout << "Input the name of stock: ";  cin >> stock[i].name;  cout << "Input the estimated earnins per share:";  cin >> stock[i].share;  cout << "Input the estimated price-to-earnins ratio:";  cin >> stock[i].ratio;  }  cout << endl;  print\_stocks(stock);  system("Pause");  return 0;  }  **Output:** |
| 3 | | #include <iostream>  using namespace std;  struct Time  {  int hours;  int minutes;  };  void next\_minutes(Time t) {  if (t.minutes == 59) {  cout << "Curent time is " << t.hours + 1 << ":" << "00" << endl;  }  else {  cout << "Curent time is " << t.hours << ":" << t.minutes << endl;  }  }  int main()  {  Time time;    cout << "Input the hour:";  cin >> time.hours;  cout << "Input the minutes:";  cin >> time.minutes;  \*/  cout << endl;  next\_minutes(time);  system("Pause");  return 0;  }  **Output:** |
| 4/6a | | #include <iostream>  using namespace std;  struct Date  {  int month;  int day;  };  void print\_next\_day(Date d) {  cout << "Curent date is " << d.month << "/" << d.day << endl;  if (d.month % 2 == 0) {  if (d.month == 2)  {  if (d.day == 28)  {  d.day = 1;  d.month += 1;  }  else {  d.day += 1;  }  }  else if (d.day == 30) {  d.month += 1;  d.day = 1;  }  else if (d.month == 12 && d.day == 31) {  d.month = 1;  d.day = 1;  }  else {  d.day += 1;  }  }  else {  if (d.day == 31)  {  d.month += 1;  d.day = 1;  }  else {  d.day += 1;  }  }  cout << "Next date is " << d.month << "/" << d.day << endl;  }  int main()  {  Date date;    cout << "Input the current month:";  cin >> date.month;  cout << "Input the current day:";  cin >> date.day;  cout << endl;  print\_next\_day(date);  system("Pause");  return 0;  }  **Output:** |
| **Array of Structures** | | |
| 5/2a | #include <iostream>  using namespace std;  const int sMonths = 12;  struct MonthDays  {  string name;  int day;  };  void display(MonthDays m) {  cout << "Name of month: " << m.name << " \nNumber of days in month: " << m.day << endl;  }  int main()  {  MonthDays convert[sMonths] = { {"January", 31}, {"February", 28}, {"March", 31}, {"April", 30},  {"May", 31}, {"June", 30}, {"July", 31}, {"August", 31}, {"September", 30},  {"October", 31}, {"November", 30}, {"December", 31} };  for(int i = 0; i < sMonths; i++)  display(convert[i]);    system("Pause");  return 0;  }  **Output:** | |
| 7/2a | #include <iostream>  using namespace std;  const int sMonths = 12;  struct MonthDays  {  string name;  int day;  };  void display(MonthDays m) {  cout << m.name << " has " << m.day << " days." << endl;  }  int main()  {  int cMonth;  MonthDays convert[sMonths] = { {"January", 31}, {"February", 28}, {"March", 31}, {"April", 30},  {"May", 31}, {"June", 30}, {"July", 31}, {"August", 31}, {"September", 30},  {"October", 31}, {"November", 30}, {"December", 31} };  cout << "Input month in numerical form[1-12]:";  cin >> cMonth;    display(convert[cMonth - 1]);    system("Pause");  return 0;  }  **Output:** | |
| 8/4a | #include <iostream>  #include <iomanip>  using namespace std;  const int number\_of\_employees = 6;  struct Employee  {  int number;  string name;  double rate;  int hours;  };  void input\_data(Employee &e) {  cout << "Input info of employee: ";  cin >> e.number >> e.name >> e.rate >> e.hours;  }  void payroll\_report(Employee e[]) {  double total\_gross\_pay = 0;  for (int i = 0; i < number\_of\_employees; i++) {  total\_gross\_pay += (e[i].rate \* e[i].hours);  cout << left << setw(20) << e[i].name << setw(20) << e[i].number << setw(20) << (e[i].rate \* e[i].hours) << endl;  }  cout << "Total gross pay:" << total\_gross\_pay << endl;  }  int main()  {  Employee list\_of\_employees[number\_of\_employees];  for (int i = 0; i < number\_of\_employees; i++)  input\_data(list\_of\_employees[i]);  cout << left << setw(20) << "Name" << setw(20) << "Number" << setw(20) << "Gross pay" << endl;    payroll\_report(list\_of\_employees);  system("Pause");  return 0;  }  **Output:** | |
| 9/5a | #include <iostream>  #include <iomanip>  using namespace std;  const int number\_of\_cars = 5;  struct Car  {  int number;  int miles;  int gallons;  };  void input\_data(Car &c) {  cout << "Input info of the car: ";  cin >> c.number >> c.miles >> c.gallons;  }  void miles\_per\_gallon(Car c[]) {  double total\_miles\_per\_gallon = 0;  for (int i = 0; i < number\_of\_cars; i++) {  total\_miles\_per\_gallon += (c[i].miles / c[i].gallons);  cout << left << setw(20) << c[i].number << setw(20) << (c[i].miles / c[i].gallons) << endl;  }  cout << "Average miles per gallon:" << total\_miles\_per\_gallon/number\_of\_cars << endl;  }  int main()  {  Car list\_of\_car[number\_of\_cars];  for (int i = 0; i < number\_of\_cars; i++)  input\_data(list\_of\_car[i]);  cout << left << setw(20) << "Number" << setw(20) << "Gallons per miles" << endl;    miles\_per\_gallon(list\_of\_car);  system("Pause");  return 0;  }  **Output:** | |
| 10/3a | #include <iostream>  using namespace std;  struct Date  {  int month;  int day;  int year;  };  int date(Date d) {    int total\_days = 0, c =0;  total\_days += (d.day - 1);  total\_days += (d.month - 1) \* 30;  total\_days += (d.year - (d.year - ((d.year - 1) % 100 + 1))) \* 360;    return total\_days;  }  int main()  {  Date currentDate;  cout << "Enter month: ";  cin >> currentDate.month;  cout << "Enter day: ";  cin >> currentDate.day;  cout << "Enter year: ";  cin >> currentDate.year;  cout << "The number of days from the turn of the century is " << date(currentDate) << endl;  system("Pause");  return 0;  }  **Output:** | |
| 11/1 | #include <iostream>  #include <cmath>  using namespace std;  struct Date  {  int month;  int day;  int year;  };  int date(Date d) {    int total\_days = 0, c =0;  total\_days += (d.day - 1);  total\_days += (d.month - 1) \* 30;  total\_days += (d.year - (d.year - ((d.year - 1) % 100 + 1))) \* 360;    return total\_days;  }  int difdays(Date a, Date b) {  return abs(date(a) - date(b));  }  int main()  {  Date firstDate, secondDate;  cout << "Enter month of first date: ";  cin >> firstDate.month;  cout << "Enter day of first date: ";  cin >> firstDate.day;  cout << "Enter year of first date: ";  cin >> firstDate.year;  cout << "Enter month of second date: ";  cin >> secondDate.month;  cout << "Enter day of second date: ";  cin >> secondDate.day;  cout << "Enter year of second date: ";  cin >> secondDate.year;  cout << "The difference between two dates is " << difdays(firstDate, secondDate) << " days" << endl;  system("Pause");  return 0;  }  **Output:** | |
| 12/1 | a)  #include <iostream>  #include <cmath>  using namespace std;  struct Date  {  int month;  int day;  int year;  };  int date(Date& d /\*This is only change we need to do\*/) {    int total\_days = 0, c =0;  total\_days += (d.day - 1);  total\_days += (d.month - 1) \* 30;  total\_days += (d.year - (d.year - ((d.year - 1) % 100 + 1))) \* 360;    return total\_days;  }  int difdays(Date a, Date b) {  return (abs(date(a) - date(b)));  }  int main()  {  Date firstDate, secondDate;  cout << "Enter month of first date: ";  cin >> firstDate.month;  cout << "Enter day of first date: ";  cin >> firstDate.day;  cout << "Enter year of first date: ";  cin >> firstDate.year;  cout << "Enter month of second date: ";  cin >> secondDate.month;  cout << "Enter day of second date: ";  cin >> secondDate.day;  cout << "Enter year of second date: ";  cin >> secondDate.year;  cout << "The difference between two dates is " << difdays(firstDate, secondDate) << " days" << endl;  system("Pause");  return 0;  }  b)  #include <iostream>  #include <cmath>  using namespace std;  struct Date  {  int month;  int day;  int year;  };  int date(Date \*d) {    int total\_days = 0, c =0;  total\_days += (d->day - 1);  total\_days += (d->month - 1) \* 30;  total\_days += (d->year - (d->year - ((d->year - 1) % 100 + 1))) \* 360;    return total\_days;  }  int difdays(Date a, Date b) {  return (abs(date(&a) - date(&b)));  }  int main()  {  Date firstDate, secondDate;  cout << "Enter month of first date: ";  cin >> firstDate.month;  cout << "Enter day of first date: ";  cin >> firstDate.day;  cout << "Enter year of first date: ";  cin >> firstDate.year;  cout << "Enter month of second date: ";  cin >> secondDate.month;  cout << "Enter day of second date: ";  cin >> secondDate.day;  cout << "Enter year of second date: ";  cin >> secondDate.year;  cout << "The difference between two dates is " << difdays(firstDate, secondDate) << " days" << endl;  system("Pause");  return 0;  } | |
| 13 | #include <iostream>  using namespace std;  struct Date {  int day;  int month;  int year;  };  Date larger(Date d1, Date d2) {  if (d2.year > d1.year) {  return d2;  }  else if (d2.year < d1.year) {  return d1;  }  else {  if (d2.month > d1.month) {  return d2;  }  else if (d2.month < d1.month) {  return d1;  }  else {  if (d2.day > d1.day) {  return d2;  }  else {  return d1;  }  }  }  }  int main() {  Date date1 = { 10, 9, 2015 };  Date date2 = { 11, 3, 2015 };  Date laterDate = larger(date1, date2);  cout << "The later date is " << laterDate.month << "/" << laterDate.day << "/" << laterDate.year << endl;  system("pause");  return 0;  }  **Output:** | |