Chapter 7 - Introduction to Classes and Objects

In the previous chapter we finished with the concept of parallel arrays which is two or more arrays where related information is found at a specific index value.

Q1: Would it be easy to sort the information in the three previously mentioned parallel arrays (id, age, and weight)?

Related information can be placed in a structure which has a general format as follows:

```
struct StructName
{
  // variable declarations
};
```

struct definition

Consider the following structure:

```
struct PayRoll
{
  int    employeeNumber;
  string name;
  double hoursWorked,
         payRate,
         grossPay;
};
```

The struct name PayRoll is a user-defined data type that can be used to declare variables. The variables that appear inside of the struct definition are members of the structure.

struct declaration

P: Declare a PayRoll variable deptHead and assign the employeeNumber, name, and payRate with the values 123, Joe Smith, and 10.00.

A: `PayRoll deptHead;`

```
deptHead.employeeNumber = 123;
deptHead.name = "Joe Smith";
deptHead.payRate = 10.00;
```

Notice: To access a member of the deptHead structure a dot is used after the variable name.

struct problem

Consider the following struct:

```
struct Time
{
  int hours,
      minutes,
      seconds;
};
```

P1: You are to write the C++ code that will read in a military time in the form hh:mm:ss and place hh into hours, mm into minutes, and ss into seconds. Error check to make sure that hh is in the range of 0-23, mm is in the range of 0-59, and seconds is in the range of 0-59.

displaying/comparing structs

Q2: Which of the following C++ statements are legal given variables time1 and time2 of type Time exist?

```
a) cout << time1 << time2;

b) if (time1 == time2)
   {
     cout << "times are equal";
   }

c) cout << time1.hours;

d) cin >> time1;

e) cin >> time1.Hours;
```

initializing a struct

```
struct Date
{
  int day,
      month,
      year;
};

Date newYears = {1, 1, 2008};

OR

Date newYears = {1, 1}; // year is undefined if unknown
```

Note: The following declaration is illegal

```
Date newYears = {1, , 2008};
```

Using a constructor

A constuctor is a special function that is called when a variable is created. The constructor name is the same as the name of the struct.

```
struct Date
{
  int day,
      month,
      year;

  Date ()
  {
    day = 1;
    month = 1;
    year = 2000;
  }
};
```

More with Constuctors

P2: Create a struct called Employee that has members name (string), age (int), gender (char). The struct is to have a constructor that initializes the name to a null string, the age to 0, and gender to F for female.

Passing arguments to constructors

It is possible to pass arguments to constructors as follows:

```
struct Date
{
  int day,
      month,
      year;

  Date (int d, int m, int y)
  {
    day = d;
    month = m;
    year = y;
  }
};
```

Q3: The declaration `Date date;` is illegal. Why?

Q4: Create a date variable equal to April 15, 2008.

passing structs to functions

structs can be passed to functions in the same way ints, float, and chars are passed to functions.

P: Consider the Date struct from the previous slide. Write a function that accepts a Date and prints the date out in the form day-month-year.

A1:

```
void printDate (Date date)
{
  cout << date.day << ‘-’ << date.month << ‘-’
       << date.year;
}
```

A2:

```
void printDate (const Date date &)
{
  cout << date.day <<‘-’ << date.month << ‘-’
       << date.year;
}
```

arrays of structs

We can declare arrays of structs. Let’s go back to the following problem:

P3: A datafile called athletes.txt exists which contains an unknown amount of information where each line of the file contains an id, age, and weight of a specific athlete. You are to write two functions as follows:

1) void readAthleteData - This function reads in up to 100 lines of data into an array of structs and returns the number of athletes in the datafile.

2) int whatAge - This function returns the age of the athlete with the given idNumber.

a)Declare a struct for the athlete’s data
b)Create an array of structs to hold all athlete’s data
c)Write each function described above