

Instructions: This homework assignment focuses on a review of the basics of C++. The questions were selected based upon the performance of the students who took the CS 1044 final exam in Fall 2000. The answers may be determined from the assigned readings in Dale/Weems/Headington and/or from the CS 1044 course notes.

Opiscans will be supplied on the first day of class and collected at the last class meeting during the first week of classes. No late opiscans will be accepted. Mutilated opiscans may be discarded.

For the following three questions, consider a C++ program consisting of three functions in a single source file, organized as follows:

```

// global includes and declarations
. . .
void f();
// definition of main()
int main() {
    // declarations of identifiers
    . . .
    // implementation code for main()
    . . .
}
// definition of f()
void f() {
    void g();
    // declarations of identifiers
    . . .
    // implementation code for f()
    . . .
}
// definition of g()
void g() {
    // declarations of identifiers
    . . .
    // implementation code for g()
    . . .
}

```

1. Placing a variable declaration before the comment "// definition of main()" results in:
 - 1) the variable being inaccessible to all the functions in the program
 - 2) the variable being "local" to main()
 - 3) an error if a variable of the same name is declared within the definition of one of the functions
 - 4) the variable being accessible to any function defined in the same file

5) 1 and 2 only	7) 3 and 4 only
6) 1, 2 and 3 only	8) None of the above

2. Placing a variable declaration within the body of main() results in:
 - 1) the variable being inaccessible to any other function unless it is passed as a reference parameter
 - 2) the variable being "local" to main()
 - 3) an error if a variable of the same name is declared within the definition of f() or g()
 - 4) the variable being accessible to any function defined in the same file

5) 1 and 2 only	7) 3 and 4 only
6) 1, 2 and 3 only	8) None of the above

For the next two questions, assume the following variable declarations:

```
bool x, y, z;
```

Consider the expression: $(!x \ || \ z \ \&\& \ !y)$

Determine the value of the boolean expression, given the indicated values for its component variables. Mark 1 if the expression would be true and 2 if the expression would be false.

14. $x == \text{false}, y == \text{true}, z == \text{false}$

15. $x == \text{true}, y == \text{false}, z == \text{true}$

For the next two questions, consider the program:

```
#include <iostream>
#include <fstream>
using namespace std;
void main() {
    ifstream iFile;
    int int1;
    char char1;
    double double1;

    iFile.open("input.dat");
    iFile >> double1;
    while (iFile ) {
        iFile.get(char1);
        iFile >> int1;
        cout << double1 << " " << char1 << " " << int1 << endl;

        iFile >> double1;
        iFile >> char1;
    }
}
```

Suppose the input file `input.dat` contains the following data, where `$` indicates the end of file character and each number is separated by a tab:

```
7      42.3  17      33      8.4   9.65$
```

16. What is the value of `char1` displayed on the second line of output?

- | | | |
|---------|--------|------------------|
| 1) '.' | 4) '2' | 7) None of these |
| 2) '\t' | 5) '3' | |
| 3) '1' | 6) '7' | |

17. What is the value of `int1` output on the third line of output?

- | | | |
|-------|-------|------------------|
| 1) 7 | 4) 17 | 7) 5 |
| 2) 42 | 5) 33 | 8) None of these |
| 3) 3 | 6) 65 | |

For the next seven questions, consider writing a program that is to read an input file of the form:

123-45-6789	Hokie, Joe Bob		
14			
Dept	Course	Credits	Grade
CS	1044	3.0	A-
CS	1205	1.0	A
CS	1104	3.0	B+
. . .			

The first line of the input file specifies a student ID and name, separated by a single tab character. The second line contains an integer, say N, specifying the number of course records to be listed, followed immediately by a newline. The third line contains column labels, which are of no significance to the program.

The next N lines are course records consisting of:

- a department abbreviation, containing no whitespace and followed immediately by a tab, then
- a four-digit course number, containing no whitespace and followed immediately by a tab, then
- a decimal specifying the number of credits, followed immediately by six spaces, then
- a letter grade consisting of 1 or 2 characters, followed immediately by a newline

The program will have the following file-scoped declarations and definitions:

```
const int MAXCOURSES = 50; // Line 1
const int MAXSTUDENTS = 10000; // Line 2

struct Course { // Line 3
    string Dept; // Line 4
    string CourseNumber; // Line 5
    double Credits; // Line 6
    string Grade; // Line 7
}; // Line 8

struct Transcript {
    string ID;
    string Name;
    int NumCourses;
    Course Courses[MAXCOURSES];
};
```

and the following array declaration in `main()` prior to any function calls:

```
Transcript Student[MAXSTUDENTS];
```

18. The programmer decides to implement a function to take the array of `Transcript` variables and initialize each cell of the array to hold easily recognizable dummy values. Why is this a good idea?
- 1) It saves time.
 - 2) It saves space.
 - 3) It makes it easier to detect certain kinds of logic errors during development.
 - 4) It prevents the occurrence of certain kinds of logic errors during development.
 - 5) All of these.
 - 5) Doing this is NOT a good idea.
 - 6) 1 and 2 only
 - 7) 3 and 4 only
 - 8) None of these

For the next five questions, consider the following design/implementation of a function to read an input file that has been formatted as described earlier. The design assumes the preconditions that the input stream has been opened on such a file, and that the Transcript variable `Stu` is an element of the array declared earlier.

```
void ReadTranscript(ifstream& iFile, Transcript& Stu) {

    int    NumRead = 0,
           NumCourses; // number of courses to be read

    Course Temp;       // temp storage for a course record during input

    // Task 1:  read student ID and name and store in the Transcript variable,
    //          and prepare for Task 2
    // Task 2:  read the number of course records
    // Task 3:  store the number of courses in the Transcript variable
    // Task 4:  prepare to read the next (first) course record
    // Task 5:  read the first course record into temporary storage

    while (iFile && (NumRead < NumCourses) ) {

        // Task 6:  store the data just read into the Transcript variable
        NumRead++;
        // Task 4:  prepare to read the next course record
        // Task 5:  read the next course record into temporary storage
    }
}
```

19. Which of the following code fragments will properly carry out Task 1 (read the student ID and name, and store them in the Transcript variable, and prepare for Task 2)?
- | | |
|---|---|
| 1) <code>iFile >> Stu.ID >> Stu.Name;</code> | 5) <code>getline(iFile, Stu.ID);
getline(iFile, Stu.Name);</code> |
| 2) <code>iFile >> Stu.ID;
iFile.ignore(1, '\t');
iFile >> Stu.Name;
iFile.ignore(1, '\n');</code> | 6) 1 and 3 only |
| 3) <code>getline(iFile, Stu.ID, '\t');
getline(iFile, Stu.Name);</code> | 7) 2 and 3 only |
| 4) <code>getline(iFile, Stu.ID, '\t');
iFile.ignore(1, '\t');
getline(iFile, Stu.Name);</code> | 8) 2 and 4 only |
| | 9) 3 and 4 only |
| | 10) None of these |
20. Assuming correct code to carry out Task 1, which of the following code fragments will then properly carry out Task 2 (actually read the number of courses)?
- | | |
|---|------------------|
| 1) <code>iFile >> NumCourses;</code> | 5) 1 and 2 only |
| 2) <code>iFile.get(NumCourses);</code> | 6) 1 and 3 only |
| 3) <code>getline(iFile, NumCourses);</code> | 7) 2 and 3 only |
| 4) All of these | 8) None of these |

21. Assuming correct code to carry out Task 1 through Task 4, which of the following code fragments will then properly carry out Task 5 (read course record into the temporary storage variables)?

- | | |
|---|---|
| 1) <code>iFile >> Temp.Dept;
iFile.ignore(1, '\t');
iFile >> Temp.CourseNumber
>> Temp.Credits
>> Temp.Grade;</code> | 4) <code>getline(iFile, Temp. Dept, '\t');
getline(iFile,
Temp.CourseNumber, '\t');
iFile >> Temp.Credits
>> Temp.Grade;</code> |
| 2) <code>getline(iFile, Temp. Dept, '\t');
getline(iFile,
Temp.CourseNumber, '\t');
iFile >> Temp.Credits;
getline(iFile, Temp.Grade);</code> | 5) All of these |
| 3) <code>iFile >> Temp. Dept;
iFile.ignore(1, '\t');
getline(iFile, Temp.CourseNumber,
'\t');
iFile >> Temp.Credits
>> Temp.Grade;</code> | 6) 1 and 3 only |
| | 7) 1 and 4 only |
| | 8) 2 and 3 only |
| | 9) 1, 3 and 4 only |
| | 10) None of these |

22. Assuming correct code to carry out Task 1 through Task 5, which of the following code fragments will then properly carry out Task 6 (store the data just read into the Transcript variable)?

- | | |
|---|--|
| 1) <code>Stu.Courses[NumRead] = Temp;</code> | |
| 2) <code>Stu.Courses[NumRead].Dept = Temp.Dept;
Stu.Courses[NumRead].CourseNumber = Temp.CourseNumber;
Stu.Courses[NumRead].Credits = Temp.Credits;
Stu.Courses[NumRead].Grade = Temp.Grade;</code> | |
| 3) Both of these | |
| 4) None of these | |

23. The course record data could be read directly into the Transcript variable. Why are temporary storage locations used in reading the course records?

- 1) It saves time.
- 2) It saves space.
- 3) It prevents possible corruption of the Course array, if an input failure occurs.
- 4) There is no good reason to do this.
- 5) None of these.

24. Referring to page 5, what does the code in lines 3 through 8 do?

- 1) Declare a global -scoped variable
- 2) Declare a function prototype
- 3) Declare a global-scoped constant
- 4) Declare a global-scoped type
- 5) None of these

For the next two questions, consider the problem of parsing input lines of the form:

```
<integer><space><character><space><integer>
```

(This might arise, for example, when parsing simple arithmetic expressions.) Assume all variables cited below are declared of appropriate types, and the file stream `In` has been opened on a file containing lines of the form given above. Assume also that the first integer value will be read by the statement: `In >> LeftOperand;`

25. Which of the following code fragments will correctly read the character value into the `char` variable `Operator`?

- | | |
|--|--------------------|
| 1) <code>In >> Operator;</code> | 5) All of these |
| 2) <code>In.get(Operator);</code> | 6) 1 and 2 only |
| 3) <code>getline(In, Operator, ' ');</code> | 7) 1 and 4 only |
| 4) <code>In.ignore(1, ' ');</code>
<code>In >> Operator;</code> | 8) 2 and 3 only |
| | 9) 1, 2 and 4 only |
| | 10) None of these |

26. Assuming that all the data through the character value has been read (but not the space after that character value), which of the following code fragments will correctly read the second integer into the `int` variable `RightOperand`?

- | | |
|--|------------------|
| 1) <code>In >> RightOperand;</code> | 5) 1 and 2 only |
| 2) <code>In.ignore(1, ' ');</code>
<code>In >> RightOperand;</code> | 6) 1 and 3 only |
| 3) <code>getline(In, RightOperand);</code> | 7) 2 and 3 only |
| 4) All of these | 8) None of these |

For the next three questions, assume the following rather uninteresting program:

```
#include <fstream>
using namespace std;

struct Point {
    double x, y;
};
const int MAXPOINTS = 100;

int ReadCorners(Point P[]);

void main() {
    Point Polygon[MAXPOINTS];

    int NumCorners;
    NumCorners = ReadCorners(Polygon);

}
// Line 1
// Line 2
// Line 3
```

The function `ReadCorners()` will read in coordinates for up to `MAXPOINTS` polygon corners, store them in the array of `Point` variables, and return the number of corners.

27. We want to add a function, called by `main()`, to the program to initialize the array of `Point` variables to hold dummy values. If the following implementation of such a function were used, with an appropriate prototype and a call added at Line 1 of `main()`, then:

```
void Init(const Point P[]) {
    Point Dummy = {0.0, 0.0};
    for (int Idx = 0; Idx < MAXPOINTS; Idx++)
        P[Idx] = Dummy;
}
```

- 1) The resulting program would not compile.
 - 2) The resulting program would compile but would not behave as specified.
 - 3) The resulting program would compile and behave as specified.
 - 4) None of these
28. We want to modify the program to apply a transformation to the coordinates of each of the `Point` variables in the array. At Line 2 of `main()`, we will add the following loop:

```
for (int Idx = 0; Idx < NumCorners; Idx++)
    flipHorizontal(Polygon[Idx]);
```

We will add the following function implementation, with an appropriate prototype, to the program:

```
void flipHorizontal(Point P) {
    P.x = -1.0*P.x;
}
```

If these changes were made, then:

- 1) The resulting program would not compile.
 - 2) The resulting program would compile but would not behave as specified.
 - 3) The resulting program would compile and behave as specified.
 - 4) None of these
29. Finally, we want to add a function that will sort the array of `Point` variables in some manner (exactly what the ordering is doesn't matter for this question). Assume that the sort function will have the prototype below, and that the body of the sort function is implemented correctly:

```
void sortCorners(Point P[], int Size);
```

Suppose we add the following call at Line 3 of `main()`: `sortCorners(Polygon, MAXPOINTS);`

Then:

- 1) The array of `Point` variables will always be sorted correctly.
 - 2) The array of `Point` variables will definitely never be sorted correctly.
 - 3) The array of `Point` variables will be sorted correctly occasionally, but not always.
 - 4) None of these
-
30. In C++, all actual function parameters, except (possibly) array names, are (by default) passed by value.
- 1) true
 - 2) false
 - 3) neither