## This exam review covers material from Chapters 1-6

1. Write the assignment statements that perform the following operations with int variable i, double variables d1 and d2, and char variable c:
   1. Add 2 to d1 and store the result in d2.
   2. Multiply d2 times 4 and store the result in d1
   3. Store the character ‘K’ in c.
   4. Store the ASCII code for the character ‘K’ in i.
   5. Subtract 1 from i and store the result back in i.

A) d2 = d1 + 2;

B) d1 = d2 \* 4;

C) c = ‘K’;

D) i = ‘K’;

E) i = i – 1;

1. Write the C++ code for a program that calculates how many days are left until Halloween, when given as an input how many weeks are left until Halloween. Use variables named weeks and days.  
   #include <iostream>

using namespace std;

int main()

{

int weeks, days;

cout << "Enter number of weeks: ";

cin >> weeks;

days = weeks \* 7;

cout << "Number of days is " << days << endl;

return 0;

}

1. Assume the following variables are defined:  
   int age;  
   double pay;  
   char section;  
   Write a single cin statement that will read input into each of these variables.

cin >> age >> pay >> section;

1. Is the following code legal? Why or why not?  
   const int DAYS\_IN\_WEEK;  
   DAYS\_IN\_WEEK = 7;

No, a named constant must be initialized at the time it is defined. It cannot be assigned a value at a later time.

1. Write a cout statement so the variable divSales is displayed in a field of eight spaces, in fixed point notation, with a decimal point and two decimal digits.

cout << fixed << showpoint << setprecision(2);

cout << setw(8) << divSales;

1. Write a cout statement so the variable profit is displayed in a field of 12 spaces in fixed-point notation, with a decimal point and four decimal digits.  
   cout << fixed << showpoint << setprecision(4);

cout << setw(12) << profit;

1. What header file must be included
   1. To perform mathematical functions like sqrt?
   2. To use cin and cout?
   3. To use stream manipulators like setprecision?  
      A) cmath B) iostream C) iomanip
2. A bowling alley is offering a prize to the bowler whose average score from bowling three games is the lowest. Write the C++ code for a program that inputs three bowling scores and calculates and displays their average.  
   #include <iostream>

using namespace std;

int main()

{

int score1, score2, score3;

float average;

cout << "Enter three scores separated by a space ";

cin >> score1 >> score2 >> score3;

average = (score1 + score2 + score3) / 3;

cout << "Average score is " << average << endl;

return 0;

}

1. What value will be stored in the variable t after each of the following statements executes?
   1. t = (12 > 1);
   2. t = (2 < 0);
   3. t = (5 == (3 \* 2));
   4. t = ( 5 == 5);  
      1, 0, 0, 1
2. Write an if/else statement that prints “Excellent” when score is 90 or higher, “Good” when score is between 80 and 89, and “Try Harder” when score is less than 80.  
   if (score >= 90)

cout << "Excellent";

else if (score >= 80)

cout << "Good";

else

cout << "Try Harder";

1. Write an if statement that sets the variable hours to 10 when the flag variable minimum is set to true.  
   if (minimum)

hours = 10;

1. Convert the following conditional expression into an if/else statement:  
   q = (x < y) ? (a+b) : (x \* 2);  
   if(x < y)

q = a + b;

else

q = x \* 2;

1. Convert the following if/else if statement into a switch statement:  
   if (choice == 1)  
   {  
    cout << fixed << showpoint << setprecision (2);  
   }  
   else if ((choice == 2) || (choice == 3))  
   {  
    cout << fixed << showpoint << setprecision (4);  
   }  
   else if ((choice == 4)

{  
 cout << fixed << showpoint << setprecision (6);  
}  
else  
{  
 cout << fixed << showpoint << setprecision (8);  
}

switch (choice)

{

case 1: cout << fixed << showpoint << setprecision(2);

break;

case 2:

case 3: cout << fixed << showpoint << setprecision(4);

break;

case 4: cout << fixed << showpoint << setprecision(6);

break;

default: cout << fixed << showpoint << setprecision(8);

break;

}

1. Write a C++ statement that prints the message “The number is valid.” If the variable grade is within the range 0 through 100.  
     
   if (grade >= 0 && grade <= 100)

cout << "The number is valid.";

1. Write a C++ statement that prints the message “The number is valid.” If the variable temperature is within the range -50 through 150.  
   if (temperature >= -50 && temperature <= 150)

cout << "The number is valid.";

1. Write a C++ statement that prints the message “The number is not valid.’ If the variable hours is outside the range 0 through 80.  
   if (hours < 0 || hours > 80)

cout << "The number is not valid.";

1. What header file do you need to include in a program that performs file operations?  
   fstream
2. What data type do you use when you want to create a file stream object that can write data to a file?  
   ofstream
3. What happens if you open an output file and the file already exists?

It will be erased and a new file with the same name will be created

1. Write C++ code that lets the user enter a number. The number should be multiplied by 2 and printed until the number exceeds 50. Use a while loop.  
   int num;

cin >> num;

num \*=2;

while (num < 50)

{ cout << num << endl;

num \*=2;

}

1. Write a do-while loop that asks the user to enter two numbers. The numbers should be added and the sum displayed. The user should be asked if he or she wishes to perform the operation again. If so, the loop should repeat; otherwise it should terminate.

double num1, num2;

char again;  
do {

cout << "Enter two numbers: ";

cin >> num1 >> num2;

cout << "Their sum is " << (num1 + num2) << endl;

cout << "Do you want to add more numbers? (Y/N) ";

cin >> again;

} while (again == 'Y' || again == 'y');

1. Write a code segment that creates an ofstream object named outfile, opens a file named tens.txt and associates it with outfile. The code should then use a for loop that writes the following set of numbers to the file, each on a new line:  
   0, 10, 20, 30, 40, 50 ….1000

It should then close the file.  
  
ofstream outfile;

outfile.open ("tens.txt");

for (int i = 0; i < 1001; i+=10)

outfile << i << endl;

outfile.close();

1. Rewrite the following code, converting the while loop to a do-while loop:  
   char doAgain = ‘y’;  
   int sum = 0;  
     
   cout << “This code will increment sum 1 or more times. \n”;  
   while (( do Again == ‘y’) || (doAgain == ‘Y”))  
   {  
    sum ++  
    cout << “Sum has been incremented. Increment again (y/n)? “;  
    cin >> doAgain;  
   }  
   cout << “Sum was incremented “ << sum << “times. \n”;

char doAgain;

int sum = 0;

cout << "This code will increment sum 1 or more times.\n";

do

{ sum++;

cout << "Sum has been incremented. "

<< "Increment it again(y/n)? ";

cin >> doAgain;

} while ((doAgain == 'y') || (doAgain == 'Y'));

cout << "Sum was incremented " << sum << " times.\n";

1. Convert the following while loop to a for loop:  
   int count = 0;  
   while (count < 50)  
   {  
    cout << “count is “ << count<< endl;  
    count++;  
   }

for (int count = 0; count < 50; count++)

cout << "count is " << count << endl;

1. Convert the following for loop to a while loop:  
   for (int x = 50; x >0; x--)  
   {  
    cout << x << “ seconds to go. \n”;  
   }  
     
   int x = 50;

while (x > 0)

{ cout << x << " seconds to go.\n";

x--;

}

1. The following statement calls a function named half which returns a value that is half that of the argument passed to it. Assume that result and number have both been defined to be double variables. Write the half function.  
     
   result = half (number);

double half(double value)

{

return value / 2;

}

1. A program contains the following function.  
   int cube (int num)  
   {  
    return num \* num;  
   }  
   Write a statement that passes the value 4 to this function and assigns its return value to the variable result.

result = cube(4);

1. Write a function named getNumber which uses a reference parameter to accept an integer argument. The function should prompt the user to enter a number in the range of 1 through 100. The input should be validated and stored in the parameter value.  
   void getNumber(int &number)

{

cout << "Enter an integer between 1 and 100): ";

cin >> number;

while (number < 1 || number > 100)

{

cout << "This value is out of the allowed range.\n"

<< "Enter an integer between 1 and 100): ";

cin >> number;

}

}

1. Write a function named biggest that receives three integer arguments and returns the largest of the three values.  
   int biggest(int num1, int num2, int num3)

{

if (num1 >= num2 && num1 >= num3)

return num1;

if (num2 >= num3)

return num2;

return num3;

}