This midterm exam was given in class several years ago. Work each of the following questions on your own. Once you are done, check your answers. For any questions whose answers you don't understand, see the TA for an explanation. This practice midterm is slightly longer than the one you can expect to see next week to provide extra questions for your practice.

### Data Types

[15 points] Examine the following C++ assignment statements variable = expression. For each statement indicate the order in which the sub-pieces are evaluated, and the values along the way. Give the value and data type (float, int, char) of the final expression as well as the value and type stored in the variable.

```c++
float Taxes;
Taxes = (20 * 4 + 30 * 3)/50;
```

Expression: Value: 3 Data Type: int

Variable: Value: 3.0 Data Type: float

```c++
float Result;
Result = 5 / 2 - 5 % 2;
```

Expression: Value: 1 Data Type: int

Variable: Value: 1.0 Data Type: float

```c++
int Distance;
Distance = 3.5 + 3 / (2 + 3);
```

Expression: Value: 3.5 Data Type: float

Variable: Value: 3 Data Type: int

```c++
int Number;\nNumber = 3 * 2 / 10.0;
```

Expression: Value: 0.6 Data Type: float

Variable: Value: 0 Data Type: int
[5 points] Rewrite the following while loop as a for loop that performs the same task.

```cpp
int Max = 1900;
int Sum = 0;\nwhile ((Max < 1950))
{
    Sum = Sum + (Max - 1900);
    cout << "Sum: " << Sum << "\n";
    Max = Max + 5;
}
```

```cpp
int Sum = 0;
for(int Max = 1900 ; Max < 1950 ; Max += 5)
{
    Sum = Sum + (Max - 1900);
    cout << "Sum: " << Sum << "\n";
}
```

[5 points] Rewrite the following do-while loop as a while loop that performs the same task.

```cpp
int Hours = 0;
int Total = 0;
do
{
    cout << "Enter a number between 0 and 20: ";
    cin >> Hours;
    if ((Hours >= 0) && (Hours <= 20))
        Total = Total + Hours;
} while (Total < 200);
```

```cpp
int Hours = 0;
int Total = 0;
cout << "Enter a number between 0 and 20: ";
cin >> Hours;
if ((Hours >= 0) && (Hours <= 20))
    Total = Total + Hours;
while(Total < 200)
{
    cout << "Enter a number between 0 and 20: ";
    cin >> Hours;
    if ((Hours >= 0) && (Hours <= 20))
```
[5 points] Rewrite the following nested for loops as a single for loop that produces the same output.

```cpp
int i, j;
for (j = 1; j < 4; j++)
{
    for (i = j; i < j+1; i++)
    {
        cout << "The value is: " << i << "\n";
    }
}
```

```cpp
for(int i=1 ;i<4 ;i++)
{
    cout << "The value is: " << i << "\n";
}
```

---

**Switch Statement**

[5 points] Write a C++ switch statement that stores 4.0 in NumberGrade if the grade is an A or a, 3.0 for B or b, 2.0 for C or c, 1.0 for D or d, and 0.0 for an F or f. It should output "invalid grade" and set value to -1.0 for any other character.

```cpp
char LetterGrade;
float NumberGrade;
cout << "Enter your grade: ";
cin >> LetterGrade;

switch(LetterGrade)
{
    case 'a':
    case 'A':
        NumberGrade = 4.0;
        break;
    case 'b':
    case 'B':
        NumberGrade = 3.0;
        break;
    case 'c':
    case 'C':
        NumberGrade = 2.0;
    case 'd':
    case 'D':
        NumberGrade = -1.0;
        break;
    case 'f':
    case 'F':
        NumberGrade = 0.0;
        break;
    default:
        cout << "invalid grade";
        NumberGrade = -1.0;
        break;
}
```
Nested Loops

[5 points] Consider the following C++ program. What does it print out? Show your work where necessary.

```cpp
#include <iostream>
using namespace std;

int main() {
    int X, Y;
    // Outer loop
    for (Y=1; Y < 10; Y = Y + X) {
        X=1;
        // Inner loop
        while(X < 3) {
            cout << "X: " << X << endl;
            X=X+1;
        }
        cout << "Y: " << Y << endl;
    }
    // Code after loop
    cout << "Final value of X: " << X << "\n";
    cout << "Final value of Y: " << Y << "\n";
    return(0);
}
```
output:
X: 1
X: 2
Y: 1
X: 1
X: 2
Y: 4
X: 1
X: 2
Y: 7
Final Value of X: 3
Final Value of Y: 10

If Statements

[10 points] Fill in a C++ program which asks the user several Y/N questions and selects a car for them based on their answers. The questions are 1) do you have children; 2) do you have lots of money; and 3) do you like trucks. The car selection should be based on the following:

Porsche: have no children, have lots of money, don't like trucks
Yukon: have no children, have lots of money, like trucks
Civic: have no children, have little money
Villager: have children, have lots of money, don't like trucks
Explorer: have children, have lots of money, like trucks
Sentra: have children, have little money

#include <iostream>
using namespace std;

int main()
{
    // Declare variables
    char Children;
    char Money;
    char LikeTrucks;

    // Get user inputs (no error checking necessary)
    cout << "Do you have children?\n";
    cin >> Children;
    cout << "Do you have lots of money?\n";
    cin >> Money;
    cout << "Do you like trucks?\n";
    cin >> LikeTrucks;
}
// Output the car choice based on info above
if (Children == 'N' && Money == 'Y' && LikeTrucks == 'N')
    cout << "nPorshe\n";
if (Children == 'N' && Money == 'Y' && LikeTrucks == 'Y')
    cout << "nYukon\n";
if (Children == 'N' && Money == 'N')
    cout << "nCivic\n";
if (Children == 'Y' && Money == 'Y' && LikeTrucks == 'N')
    cout << "nVillager\n";
if (Children == 'Y' && Money == 'Y' && LikeTrucks == 'Y')
    cout << "nExplorer\n";
if (Children == 'Y' && Money == 'N')
    cout << "nSentra\n";

return 0;
[10 points] For each of the following function prototypes, which of the following calls are syntactically correct? Note: function calls that would compile with warnings or compile without warnings but cause implicit type conversions should be considered incorrect.

```c
// Variable declarations
int A, B, C;
float X, Y;
char Char;

// Function prototypes
int Maximum(const int Num1, const int Num2);
void Total(const float A, const float B, float &C);
char GetChar();
```

```c
Maximum (A,B); Correct(Y or N):__________
A = Maximum (7,3); Correct(Y or N):__________
A = Maximum (Num1, Num2); Correct(Y or N):__________
Total (A, B, C); Correct(Y or N):__________
Total (3.0, X, Y); Correct(Y or N):__________
Total (3.0, 5.0, 8.0); Correct(Y or N):__________
Total (Y, X, Y); Correct(Y or N):__________
GetChar (Char); Correct(Y or N):__________
Char = GetChar (); Correct(Y or N):__________
```
Defining Functions

[10 points] For this question, your task is to write function prototypes (the first line of the function that gives the return type and the list of parameters). You do NOT have to fill in the function bodies. Use const where appropriate.

// This function takes in two numbers and returns their average.

float Average ( const float A, const float B );
OR
void Average ( const float A, const float B, float &Average );

// This function asks the user how many sisters they have and returns their answer.

int GetSisters ( );
OR
void GetSisters(int &Answer);

// This function takes in one number and returns the square root of this number.

float SquareRoot (const float N);
OR
void SquareRoot (const float N, float &sqrtN);

// This function take in a letter grade and returns
// the corresponding grade point value (eg A = 4.0).

float GetGPA (const char Letter);
OR
void GetGPA (float &GPA, const char Letter);
```cpp
#include <iostream>
using namespace std;

const int X = 3;

void Proc1 (int &Y)
{
    Y = 6;
    cout << "Inside Proc1, X: " << X << " Y: " << Y << "\n";
}

int Proc2 (int &Y, const int Z)
{
    int X = 7;
    Proc1 (X);
    Proc1 (Y);
    Y = 3;
    cout << "Inside Proc2, X: " << X << " Y: " << Y << " Z: " << Z << "\n";
    return (Y + Z);
}

int main()
{
    int A = 1;
    int B = 3;
    int C = 5;

    C = Proc2 (A,B);
    cout << "Finally, A: " << A << " B: " << B << " C: " << C << "\n";
    return(0);
}
```
[5 points] Using the box method, trace the execution of the program above to show the sequence of function calls, the values of variables, and return values.

```
X = 3
```

```
main
A = y, B: 3
B: 3
C = B: 6
```

```
proc2 (A, B)
y = A
z = 3
x = y: 6
y = 3
```

```
proc1 (x)
y = x
y = 6
```

```
proc1 (y)
y = y
y = 6
```

[5 points] Using this information, what does the program output?

---

```
Inside Proc1, X: 3 Y: 6
Inside Proc1, X: 3 Y: 6
Inside Proc2, X: 6 Y: 3 Z: 3
Finally, A: 3 B: 3 C: 6
```