# Summer 2018 CS320 Midterm 1

#### Section A – Multiple Choice & True/False ( 6pts )

1. Which best describes imperative programming?
   1. **Languages where the programmer specifies how to solve problems.**
   2. Languages where the programmer specifies which problem to solve.
   3. Languages that have control structures like if/else and for.
   4. Languages that allow for calling other portions of code.
2. Why do we want to allocate memory on the stack?
   1. **Memory allocated on the stack has a lifetime limited to the function that declares it.**
   2. Pointers to memory on the stack cannot be passed to other functions.
   3. Memory on the heap doesn’t require additional memory management.
   4. Memory on the stack must be freed otherwise there is a memory leak.
3. Which best describes strictly typed languages?
   1. A language where only certain kinds of data can be stored in variables.
   2. Languages that do not allow creation of new data types.
   3. Languages where data contains the typing information.
   4. **Languages where variables contain the type typing information.**

#### Section B – Short Answer (25 words or less for each) ( 15pts )

1. When would using a declarative language be useful (one good situation is enough)?  
   **Query a database.**
2. Why must programmers understand how scope works in the languages they are using?  
     
   **Programmers must know which value is used by name in each scope.**
3. In what situations is it incorrect to return a pointer to a piece of data?  
   **When the data was declared in currently reducing function.**
4. What do data types do (i.e. what is the purpose of a data type)?   
   **Dictates how much memory to allocate.**

**Specifies how data should be interpreted**.

1. Describe the difference between early and late binding.

#### Section C – Fill in the Blanks (10pts )

Write you answer **IN THE BOXES PROVIDED.**

1. How much memory is allocated on the stack when the following function is called?

int foo(int x, int y){

char buff[256];

printf("Hello World\n");

}

**264**

1. What does the following print to STDOUT?

int foo(int n){

if(n<2)

return n;

else

return foo(n-2) + foo(n-1);

}

int main(int argc,char \*argv[]){

int x = (int)3.14f;  
 printf("%d",foo(x));

return 0;

}

**2**

#### Section D – Program ( 10 pts )

In C, solve the following problem.

1. Write a program that prints the sum of numbers 1-N, where N is a number the user types on STDIN.

Int main(){

Int N;

Scanf(“%d”, N};

Int sum =0;

For(int i=1; i<= N; I++){

Sum += I;

}

Printf(“%d”, sum);

}