Christian Justin J. Salinas
Instructor Ara Abigail Ambita
CMSC 21 – 2
Lecture 1 Assignments

1.

```c
#include <stdio.h>

int main(void) {
    printf("a.  In C, lowercase letters are significant.\n");
    printf("b.  main is where program execution begins.\n");
    printf("c.  Opening and closing braces enclose program statements in a routine.\n");
    printf("d.  All program statements must be terminated by a semicolon.\n");

    return 0;
}
```

2.

```c
#include <stdio.h>
int main (void){
printf ("Testing...");
printf ("....1");
printf ("...2");
printf ("..3");
printf ("\n");
return 0;
}
```

The program above outputs the following:

```
Testing.......1...2..3
```

3.

```c
#include <stdio.h>

int main(void) {
    // Start of Variable Declaration and Initialization
    int value1, value2, result;
    value1 = 87;
    value2 = 15;
    // End of Variable Declaration and Initialization

    // Assign subtraction of two values
    result = value1 - value2;

    // Display output by format string
    printf("%d - %d is %d", value1, value2, result);

    // Indicate success of execution
    return 0;
}
```

4.

```
1  #include <stdio.h>
2  int main(Void)
3  INT sum;
4  /* COMPUTE RESULT
5  sum = 25 + 37 - 19
6  /* DISPLAY RESULTS //
7  printf ("The answer is %i\n" sum);
8  return 0;
9  }
10
```

Syntactic errors:

Line	Reason	Fix
2	• No opening brace "{"	Add "{".
3	• Error: unknown type name 'INT'. • It's best practice to keep data type declarations in lowercase.	Change "INT" to "int".
4	• Missing closing tag for comment	Add "*/" at the end.
5	• Missing semicolon	Add ";" at the end.
6	• Comment tags do not match	"//" can be used as a one-liner comment. Replacing "//" to "*/" also works.
7	• No comma after string format	Add "," before sum

For best practice:

Line	Reason	Fix
2	• C is a case-sensitive language. • Writing void instead of just leaving function arguments empty is not required at all (old syntax).	Change "V" to "v".
7	• Whitespace after printf	Remove whitespace
7	• %d and %i behave similar with printf, however, they differ in scanf.	Use %d when expecting a decimal integer. Use %i when expecting 0 and 0x prefixes as octal and hexadecimal.
3-8	• Proper indentation for better readability	Add indentation

Corrected program:

```
1  #include <stdio.h>
2  int main(void) {
3      int sum;
4      /* COMPUTE RESULT */
5      sum = 25 + 37 - 19;
6      /* DISPLAY RESULTS */
7      printf ("The answer is %d\n", sum);
8      return 0;
9  }
10
```

5.

```
1  #include <stdio.h>
2  int main (void){
3  int answer, result;
4  answer = 100.
5  result = answer - 10;
6  printf ("The result is %i\n", result + 5);
7  return 0;
8
```

The program above will not run and has the following errors:

Line	Reason	Fix
4	error: expected ';' before 'result'	Replace '.' with a semicolon
7	error: expected declaration or statement at end of input	Add '}' after return 0;

Corrected program:

```
1  #include <stdio.h>
2  int main (void){
3      int answer, result;
4      answer = 100;
5      result = answer - 10;
6      printf ("The result is %d\n", result + 5);
7      return 0;
8  }
9
```

After applying the fixes, the program outputs:

```
The result is 95
```