

This print-out should have 18 questions. Multiple-choice questions may continue on the next column or page – find all choices before making your selection. The due time is Central time.

EE345L Valvano Homework 2.

---

**001** (part 1 of 1) 10 points

In the following C program, what is the effect of the **static** modifier?

```
short AA;
void function(void){
    static short BB;
}
```

1. The **static** modifier changes the situation so the variable is now permanently allocated in memory, but the variable is still accessible only within this function.
2. None of these choices is correct.
3. The **static** modifier changes the situation so the variable is read-only.
4. The **static** modifier changes the situation so the program runs more efficiently.
5. The **static** modifier changes the situation so the variable is now only accessible within the scope of the function.

---

**002** (part 1 of 1) 10 points

In the following C program, what is the effect of the **static** modifier?

```
static short AA;
void function(void){
    short BB;
}
```

1. The **static** modifier changes the situation so the variable is now only accessible within the scope of this file.
2. The **static** modifier changes the situation so the variable is read-only.
3. None of these choices is correct.

4. The **static** modifier changes the situation so the variable is now permanently allocated in memory.

5. The **static** modifier changes the situation so the program runs more efficiently.

---

**003** (part 1 of 1) 10 points

In the following C program, what is the effect of the **static** modifier?

```
short AA;
void static function(void){
    short BB;
}
```

1. The **static** modifier changes the situation so the function is read-only.
2. The **static** modifier changes the situation so the function is now only accessible within the scope of this file.
3. None of these choices is correct.
4. The **static** modifier changes the situation so the program runs more efficiently.
5. The **static** modifier changes the situation so the function is now permanently allocated in memory.

---

**004** (part 1 of 1) 10 points

In the following C program,

```
short AA=4;
const short BB=5;
void function(void){
    short CC;
    for(CC=1;CC<10;CC++){
        short DD;
    }
}
```

which variables are local?

1. Just AA.
2. Just CC.

3. CC and DD.
4. Just BB.
5. BB and CC.
6. AA and BB.
7. Just DD.
8. None of these choices is correct.

---

**005** (part 1 of 1) 10 points

In the following C program, what is the effect of the **const** modifier?

```
short AA;
void function(const short BB;){
}

```

1. The **const** modifier changes the situation so the parameter is read-only.
2. The **const** modifier changes the situation so the parameter is now only accessible within the scope of the function.
3. None of these choices is correct.
4. The **const** modifier changes the situation so the parameter is now permanently allocated in memory.
5. The **const** modifier changes the situation so the program runs more efficiently.

---

**006** (part 1 of 1) 10 points

In the following C program, what is the effect of the **const** modifier?

```
const short AA=5;
void function(void){
    short BB;
}

```

1. The **const** modifier changes the situation so the program runs more efficiently.
2. The **const** modifier changes the situation

so the variable is now only accessible within the scope of this file.

3. The **const** modifier changes the situation so the variable is now permanently allocated in memory.
4. None of these choices is correct.
5. The **const** modifier changes the situation so the variable is read-only.

---

**007** (part 1 of 5) 5 points

Consider the following C program, which is implemented on an embedded system. **Global** RAM means permanently allocated at a fixed location in volatile memory. **Stack** RAM means temporarily allocated, used, then deallocated in volatile memory. **EEPROM** means permanently allocated at a fixed location in nonvolatile memory. Please note that the variable names in this example do not follow the standard naming conventions.

```
const short aa=1000;
static short bb=1000;
short add3(const short cc){
    static short dd;
    dd = aa+cc;
    return(dd);
}
void main(void){
    short ee;
    ee = add3(bb);
}

```

Where in memory is the variable **aa** stored?

1. global RAM
2. EEPROM
3. stack RAM

---

**008** (part 2 of 5) 5 points

Where in memory is the variable **bb** stored?

1. global RAM
2. EEPROM

**3.** stack RAM

---

**009** (part 3 of 5) 5 pointsWhere in memory is the variable `cc` stored?

1. stack RAM
2. EEPROM
3. global RAM

---

**010** (part 4 of 5) 5 pointsWhere in memory is the variable `dd` stored?

1. EEPROM
2. global RAM
3. stack RAM

---

**011** (part 5 of 5) 5 pointsWhere in memory is the variable `ee` stored?

1. EEPROM
2. global RAM
3. stack RAM

---

**012** (part 1 of 1) 10 points

Consider the situation where a 7 bit parameter is added to a 15 bit parameter. In particular, assume the number  $n1$  can be any value from 0 to 127, and the number  $n2$  can be any value from 0 to 32767.

If  $n3 = n2 + n1$ , how many bits are required to store the number  $n3$ ?

Answer in units of bits.

---

**013** (part 1 of 1) 10 points

Consider the situation where a 6 bit parameter is multiplied by a 11 bit parameter. In particular, assume the number  $n1$  can be any value from 0 to 63, and the number  $n2$  can be any value from 0 to 2047.

If  $n3 = n2 * n1$ , how many bits are required to store the number  $n3$ ?

Answer in units of bits.

---

**014** (part 1 of 1) 5 points

According to the standard naming convention, what type is the variable `Error_Count`?

1. Local variable
2. Public global variable
3. Private global variable
4. A compile-time constant, not stored in memory

---

**015** (part 1 of 1) 5 points

According to the standard naming convention, what type is the variable `ErrorCount`?

1. Private global variable
2. Local variable
3. A compile-time constant, not stored in memory
4. Public global variable

---

**016** (part 1 of 1) 5 points

According to the standard naming convention, what type is the variable `errorCount`?

1. Public global variable
2. Local variable
3. Private global variable
4. A compile-time constant, not stored in memory

---

**017** (part 1 of 1) 5 points

According to the standard naming convention, what type is the object named `ERROR_COUNT`?

1. Private global variable

**2.** A compile-time constant, not stored in memory

**3.** Local variable

**4.** Public global variable

---

**018** (part 1 of 1) 5 points

According to the standard naming convention, what is the type of the following function?

```
short Error_Count(void){
    static short count=0;
    count++;
    return(count);
}
```

**1.** Private function

**2.** Public function

**3.** Local function