COSC1076 | Advanced Programming Techniques

Revision Questions | Week 03

These are self-revision questions, to help you track if you are understanding the weekly course content.

You should FIRST answer these questions using "pen-and-paper". Only after this should you test your answers by writing and compiling programs.

1. What is the purpose of a `typedef`?
2. Create a `typedef` for a pointer to a `double`.
3. Use the following Class description and answer the following questions:

```
class Example {
public:
    Example(int value);
    ~Example();

    void noParams();
    void someParams(double value);
};
```

   (a) What is the purpose of the `~Example()` method?
   (b) Write code to create a variable of the `Example` class, that calls the constructor with the value 8.
   (c) Write code to create a pointer to the above variable.
   (d) Using the direct variable, write code to call the `noParams()` method.
   (e) Using the pointer, write code to call the `someParams()` method.
4. "Under-the-hood", what does C/C++ use to manage arrays?
5. What are the two ways to interpret the following type?

```
double** dbl;
```

6. Using the code snippet, answer the following questions:

```
double* array;
array[0] = 1.2;
```

   (a) What is the problem with the code snippet?
   (b) Fix the error, using *two* different methods.
7. Given the following code, which components should be placed in a header file, and which should be placed in a code file?

```
#include <iostream>

#define EXIT_SUCCESS    0
#define LENGTH          5

using std::cout;
using std::endl;

int foo(int x);

class Example {
public:
    Example(int value);
    ~Example();

    void noParams();
```

```
};

int void main(void) {
    Example ex(1);
    ex.noParams();

    return EXIT_SUCCESS;
}

int foo(int x) {
   int y = 3;
   return x + y;
}

Example::Example(int value) {
}

Example::~Example() {
}

void noParams() {
}
```

8. What are the three types of memory locations used by a typical program?
9. What is the purpose of each of these types of memory locations?
10. For the below program, draw the changes to the call stack.

```
#define EXIT_SUCCESS    0

void fnA(int* value);
int fnB(int dbl);

int main (void) {

   int a = 7;
   int b = 8;

   a = 10;
   fnA(&b);

   return EXIT_SUCCESS;
}

void fnA(int* value) {
   int tmp = 0;
   tmp = dnB(*value);
   *value = tmp;
}

int fnB(int dbl) {
   int calc = 0;
   calc = dbl * dbl;
   return cal;
}
```

11. Write a C++ code snippet to:
    (a) Create a new `double` on the heap
    (b) Create an array of characters on the heap, and set the value to `"hello world!"`;
    (c) Delete the `double`

(d) Delete the array of characters;
(e) Create an object of the `Example` class in question 3 on the heap
(f) Delete the `Example` class

12. Look at the below program that implements a class, and answer the following questions:

```
class ContainsError {
public:
    ContainsError(int value);
    ~ContainsError();

private:
    int* ptr;
};

ContainsError::ContainsError(int value) {
    ptr = new int(value);
}

ContainsError::~ContainsError() {
}
```

(a) What is the error in this code?
(b) Fix the error.