A QUICK Q & A

Q1

- `1) x = pow(y,2.0);`
- `2) x = y*y;`

- `3) a=c/2.0;`
- `4) a=c*0.5;`

- `5) int var, y = 10;  var = (y<10) ? 30:40;`

1. Which implementation is more efficient, 1) or 2)?
2. Which implementation is more efficient, 3) or 4)?
3. What is the result of 5)?

"? :" is called conditional operator

- Exp1 ? Exp2 : Exp3;
- The value of a ? expression is determined like this: Exp1 is evaluated. If it is true, then Exp2 is evaluated and becomes the value of the entire ? expression. If Exp1 is false, then Exp3 is evaluated and its value becomes the value of the expression.
- Another implementation which has the same result:

```
if(y < 10)
{
    var = 30;
}
else
{
    var = 40;
}
```

Q2

- What can go wrong if the following block of code is in a program?

```
int iarray[20], i;

for(i=0;i <= 20; i++) {
    iarray[i] = 0;
}
```

Q2

```
struct address {
int number;
char *street;
char state[2];
int zip;
};

…
address jd, *pjd;
pjd = &jd;
pjd->number = 60;
```

1. What does the keyword "`struct`" mean?
2. What data type is "`pjd`"?
3. What is "`->`" called and what does it do?

Q3

- What is dynamic memory? Give an example when dynamic memory is needed.
- What is the difference between dynamic memory and memory allocated to normal variables “int a;”.
- Can you write a function which allocate computer memory storing a M×N size two-D array of data dynamically?
  - In C, use `malloc()`
  - In C++, use `new`

Q4

```
#if !defined(_STD_LIB_H)
#define _STD_LIB_H

#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <new>
#include <string>

extern double addition(double, double);

#endif
```

1. What is "`_STD_LIB_H`" called?
2. What are "`#if !defined(_STD_LIB_H)`","`#define _STD_LIB_H`", "`#endif`" called?
3. What is the purpose of having "`if !defined(_STD_LIB_H)`", "`#endif`"
4. What does "`extern`" do to the definition?

see http://www.cplusplus.com/doc/tutorial/preprocessor/

- https://gcc.gnu.org/onlinedocs/cpp/Macros.html
  - A *macro* is a fragment of code which has been given a name. Whenever the name is used, it is replaced by the contents of the macro. There are two kinds of macros. They differ mostly in what they look like when they are used. *Object-like* macros resemble data objects when used, *function-like* macros resemble function calls.

- https://www.tutorialspoint.com/cplusplus/cpp_preprocessor.htm
  - The preprocessors are the directives, which give instructions to the compiler to preprocess the information before actual compilation starts.

Q5

```
#include<iostream>
#include<stdio.h>
using namespace std;

class Test
{
    public:
    Test() {}
    Test(const Test &t)  // what is this function called?
    {
        cout<<"Copy constructor called "<<endl;
    }

    Test& operator = (const Test &t)  //what is this called?
    {
        cout<<"Assignment operator called "<<endl;
        return *this;
    }
};

int main()
{
    Test t1, t2;
    t2 = t1;
    Test t3 = t1;
    return 0;
}
```

https://www.geeksforgeeks.org/copy-constructor-vs-assignment-operator-in-c/

Q6

Which of the followings is/are automatically added to every C++ class, if we do not write our own.

(A) Copy Constructor

(B) Assignment Operator

(C) A constructor without any parameter

Computational Thinking (CT)

- Jeannette Wing

computational thinking (CT) is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could also execute.

https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf