
1.124J Foundations of Software Engineering

Problem Set 1

Due Date: Tuesday 9/19/00

Reference Readings: *From C++ Primer:*

- *Chapter 1 and Chapter 2 (only skim through)*
- *Chapters 3, 4, 5 and 6*

Problem 1:[15%]

This is a multiple choice question whose purpose is to expose you to several detail issues of C++ programming that are related to a variable definition, data types, mixed expressions, conversions, etc. You need to select the correct answer(s), or provide the answer, as indicated in the following questions. There may be one or more than one correct answers in the multiple choice questions. Please submit this page, stapled together with the hardcopies for the other problems, with the selected answer(s) to each question circled. Please, write your name and username clearly on this page as well. Each question counts for 1 point.

Questions 1-3: Consider the following code:

```
#include <iostream.h>
extern int x;           // Statement a
int main()
{
    int a, b;          // Statement b
    double d = 9;     // Statement c
    a = 1;             // Statement d
    b = 5;             // Statement e
    cout << x+ a/b + d/10 << endl;
}
```

1. Which line of the above code is both a definition and an initialization?

2. Which line of the above code declares a variable without allocating memory for it?

3. Which lines of the above code are assignments?

4. According to the following statement: `const int *p;`

- a. the value of the pointer p cannot change
- b. the value of the integer that p points to cannot change
- c. both pointer p and the value of the integer that p points to cannot change
- d. both pointer p and the value of the integer that p points to can change
- e. p is a constant pointer to int

5. When the following logical test is true?

`(x >= y && !x && x * y < 0 && y == 0)`

- a. if x is greater than y, and y is equal to zero
- b. if both x and y are equal to zero
- c. if x is positive, and y is equal to zero
- d. always
- e. never

6. Which of the following cases of mixed expressions is/are correct (circle the correct one(s)), considering the following definition:

`double d; float f; int i; char c;`

- a. `'f' - 'd'` is a double
- b. `f / 3.33` is a float
- c. `'f' - 'd'` is an int
- d. `'f' - 'd'` is a char
- e. none of the above

7. What is the result of the statement following the definitions given below?

```
char c='b';  
char *pc=&c;  
char *&rc=pc;  
(*rc)++;
```

- a. it increases $\&rc$
- b. it stores 'b' in variable c
- c. it increases $*rc$
- d. it increases pc , by one byte
- e. none of the above

8. Considering the following definitions, which of the provided statements (if any) are invalid?

```
double x= 0.5, y=4.9;
double *px, *py, &rx=x;
```

- a. $px = \&x$; $\text{double } \&rx = *px$;
- b. $px = py$;
- c. $\text{double } \&ry = rx$;
- d. $px = px = rx$;
- e. $px = py = *x$;

9. Considering the following definitions, which of the provided statements (if any), would give the value of x , assuming that x is a double that has been properly defined and initialized to a value?

```
void *pp = &x;
double *px = &x;
double **ppx = &px;
```

- a. $**ppx$
- b. $*(\text{static_cast } \langle \text{double}^* \rangle (pp))$
- c. $*pp$
- d. $*(\&px)$
- e. $*((\text{double}^*)pp)$

10. Which of the following expressions give(s) as result an *int* equal to 6?

- a. 'z' - 't'
- b. $13 \% 7$
- c. $7 \% 2$
- d. $29/5$
- e. $55 \% 7$

11. What will be the value of x after the execution of the following line?

`int x = (7>6 ? 1+8 : 8)`

- a. 6
- b. 7
- c. 1
- d. 8
- e. 9

12. Which of the following(s) is a (are) valid function declaration (i.e. prototype)?

- a. `void func(int x);`
- b. `void func(int){};`
- c. `func(9.5);`
- d. `void func(int x){ // Function body };`
- e. `void func(int);`

13. Which of the following functions, whose declarations are given below, will be called:

```
float f;  
printFun(2.0*f);
```

- a. `void printFun(void)`
- b. `void printFun(double)`
- c. `void printFun(float)`
- d. `float printFun(float)`
- e. none of the above

14. How many times is function *fib* called when *num* is 3, including the initial *fib*(3)?

```
int fib(int num) // Fibonacci value of a number  
{  
    switch(num)  
    {  
        case 0:  
            return(0);  
            break;  
        case 1:  
            return(1);  
            break;  
        default:  
            return(fib(num - 1) + fib(num - 2));  
    }  
}
```

```
        break;
    }
}
```

- a. 1
- b. 4
- c. 3
- d. 5
- e. none

15. In a function with return type *void*, what happens at return?

- a. 0 is returned
- b. 1 is returned
- c. An arbitrary integer is returned
- d. A void pointer is returned
- e. No value is returned

Problem 2:[5%]

Given the definitions of the variables below, determine the data type of the following expressions:

```
bool b;
char c;
int i;
float f;
double d;
```

1. $77 + c + i + 1L$
2. $6.55f + f / 1.5 - 9 / 8$
3. $'z' - 'z'$
4. $b + c$
5. $'t' - 'a' + c$
6. $77.8f + 4 * 0.5f + 45L$
7. $42L + (\text{int}) d + 94.3f + \text{int}(4.9)$
8. $0.0 + f + c$
9. $5.28L * d * 3 + 4.5$

10. $1.5f / d * f + 6.9 * 4L$

Problem 3:[10%]

The following program uses a local and a global array of characters with the same name, called *name*. There is also an object of the class *Material* that has a member called *name*. The declarations of the functions and the *Material* class are provided in the file *ps1_3.h* and the definitions and *main()* are provided in the file *ps1_3.C*. The only thing that you need to do is to replace the 3 comments with the proper expressions, so as to print out the local variable (in *main()*) called *name*, the global variable *name* and the member *name* of the object *m*. Compile and execute the program and submit both the printout of the modified code and a screendump of the output. Also electronically turn-in the source code file and the makefile you have used for the compiling.

ps1_3.h

// Problem Set#1 - Problem#3 [ps1_3.h]

```
#ifndef PS1_3_H
#define PS1_3_H

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

class Material
{
public:
    char name[20];
    double modulusElasticity;
    double ratioPoisson;
    Material();
    void print(void);
};

#endif
```

ps1_3.C

// Problem Set#1 - Problem#3 [ps1_3.C]

```
#include "ps1_3.h"
```

```
Material::Material()
```

```
{  
    strcpy(name, "None");  
    modulusElasticity = 0.0;  
    this -> ratioPoisson = 0.0;  
}
```

```
void Material:: print(void)
```

```
{  
    cout << "\n Material: " << name  
        << "\n Modulus of elasticity = " << modulusElasticity  
        << "\n Poisson ratio = " << ratioPoisson << endl;  
}
```

```
char name[40] = "Foundation of Software Engineering";
```

```
int main()
```

```
{  
    char name[30] = "Problem 3";  
    Material m;
```

```
    cout << "\n Local name : "  
        << "replace this comment with the expression to print the local name";  
    cout << "\n Global name : "  
        << "replace this comment with the expression to print the global name";  
    cout << "\n Object name : "  
        << "replace this comment with the expression to print the name of object m";
```

```
    cout << "\n Object: " ;
```

```
    /* replace this comment with a statement that invokes  
       the member function print() using the object m */
```

```
    cout << "\n Exiting properly\n" << endl;  
    return EXIT_SUCCESS;  
}
```

Problem 4:[10%]

What the following sets of statements output?

1.

```
int x;  
int *y;  
int **z;  
  
x = 0;  
y = &x;  
z = &y;  
  
x++;  
(*y)++;  
(**z) = 10;  
cout << "x is " << x <<  
" *y is " << *y <<  
" **z is " << **z << endl;
```

2.

```
void increment (int &a, int b, int *c)  
{  
    a++;  
    b++;  
    (*c)++;  
    cout << "a is " << a << " b is " << b << " c is " << *c << endl;  
}  
  
int main()  
{  
    int x=0, y=0, z=0;  
    increment (x,y,&z);  
    cout << "x is " << x << " y is " << y << " z is " << z << endl;  
}
```

3.

```

void swap(int *a, int *b)
{
    int *tmp;

    tmp = a;
    a = b;
    b = tmp;
}

int main (){
    int x[] = {1, 1, 1};
    int y[] = {2, 2, 2};
    swap (x,y);
    cout << " x[0] = " << x[0] << " y[0] = " << y[0] << endl ;
}

```

4.

```

double scale(double x, double s=1, double offset=0);

```

```

double scale(double x, double s, double offset)
{
    return s*x + offset;
}

```

```

int main()
{
    double value=20;

    double test1 = scale(value);
    double test2 = scale(value,2.54);
    double test3 = scale(value, 9.0/5.0, 32.0);

    cout << "test1=" << test1
         << "test2=" << test2
         << "test3=" << test3 << endl;
    return 0;
}

```

5.

```
void func2 (int& a, int& b)
{
    int tmp = a;
    a = b;
    b = tmp;
}
```

```
void func1 (int a, int b)
{
    a /= 2;
    b *= 2;

    cout << "Before func2" << endl;
    cout << " a = " << a << " b = " << b << endl;

    func2 (a, b);

    cout << "After func2" << endl;
    cout << " a = " << a << " b = " << b << endl;
}
```

```
int main()
{
    int a = 10;
    int b = 20;

    cout << "At the beginning" << endl;
    cout << " a = " << a << " b = " << b << endl;

    func1 (a, b);
    func2 (a, b);

    cout << "At the end" << endl;
    cout << " a = " << a << " b = " << b << endl;
}
```

Problem 5:[40%]

For this problem you need to write a program that prompts the user for an odd positive integer, n , greater than 1, and then prints the following hourglass figure which has a height of n text lines:

In this example: $n = 9$

```
*****
*      *
*      *
*  *
*
*  *
*      *
*      *
*****
```

Your program should check that the user enters a valid number for n and print an error message if not. Your output should look exactly like the figure above: symmetric and left-justified, with all `*'s and spaces in the locations as shown. It should be scaled appropriately for the given value of n . Please make sure that your program works for all valid cases (e.g. check for $n=3$). It is recommended to break the entire job into small tasks to handle each region of the figure.

Please, provide your source code in a file named *ps1_5.C* and any declarations in the corresponding header file *ps1_5.h*.

Problem 6:[20%]

In this problem you should provide the missing constructor in *ps1_6.h* which should serve as both the default (to zero both member variables) and as one with two doubles (named *real* and *imaginary*) as arguments (to set the member variables to the values of the corresponding arguments).

Also, you also need to provide in the file *ps1_6.C* the missing definitions of the member functions *double get_real(void)*, *double get_imaginary(void)*, *void set_real(double)* and *void set_imaginary(double)*.

Please, compile and execute the completed program and submit printouts of both the source code and the output (e.g. screen dump).

ps1_6.h

```
#include <iostream.h> // Problem Set#1 - Problem#6 [ps1_6.h]
#include <stdlib.h>

class Complex
{
private:
    double real;
    double imaginary;

public:

    /*      Provide the body of ONLY one constructor which can serve
    both as the default constructor, (setting the values of both member
    variables to 0.0), and as a constructor with arguments two doubles
    named real and imaginary, and which should initialize the real and
    imaginary member variables to the provided parameters real and
    imaginary (local variables for the constructor)          */

    double get_real(void);
    double get_imaginary(void);
    void set_real(double);
    void set_imaginary(double);

};
```

ps1_6.C

```
#include "ps1_6.h" // Problem Set#1 - Problem#6 [ps1_6.C]

/*      Provide the definitions of the member
    functions get_real() & get_imaginary() */

/*      Provide the definitions of the member
    functions set_real() & set_imaginary() */
```

```
int main ( )
{
    Complex c1;

    cout << "\n\n c1 = " << c1.get_real()
        << " + " << c1.get_imaginary() << "i " << endl ;

    Complex c2(7.25,-8.5);
    cout << "\n\n c2 = " << c2.get_real()
        << " + " << c2.get_imaginary() << "i " << endl ;

    c1.set_real(1.7);
    c1.set_imaginary(-6.7);
    cout << "\n\n c1 = " << c1.get_real()
        << " + " << c1.get_imaginary() << "i " << endl << endl;

    return EXIT_SUCCESS ;
}
```

Note:

Please submit **both** printouts of the source code you have written (preferably using `% enscript -2Gr -Pprinter filename`) and screen dumps of the execution output (using `%xdpr -Pprinter`), with your name and username clearly written on the first page of the **stapled** submitted problem set. The submitted code must be identical to that electronically turned in (as described above).

If you can submit the problem set one day late 10% of the total grade (i.e. 10 points) will be deducted. More than one day late problem sets will **not** be accepted.
