Problem 1: [15%]

Questions 1-3: Consider the following code:

```
#include <iostream.h>
extern int x;
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?
   -> C

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

3. Which lines of the above code are assignments?
   -> d & e

4. According to the following statement: const *int p;
   
   b. the value of the integer that p points to cannot change
5. When the following logical test is true?

\[ (x \geq y \land \neg x \land x \cdot y < 0 \land y = 0) \]

e. never

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

\[ \text{double } d; \text{ float } f; \text{ int } i; \text{ char } c; \]

c. 'f' - 'd' is an int

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

\[
\text{char } c='b'; \\
\text{char } *pc=&c; \\
\text{char } *&rc=pc; \\
(*rc)++; \\
\]

c. it increases *rc

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

\[
\text{double } x= 0.5, y=4.9; \\
\text{double } *px, *py, &rx=x; \\
\]

a. px = &x; double &rx = *px ;

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?

\[
\text{void } *pp = &x; \]
double *px=&x;
double **ppx=&px;

a. **ppx   
b. *((static_cast <double*>(pp))  
d. *((&px))   
e. *((double*)pp)

10. Which of the following expressions give(s) as result an int equal to 6?
   a. ’z’ - ’t’   
   b. 13 % 7   
   c.  55 % 7

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

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

   e. 9

12. Which of the following is a valid function declaration (i.e. prototype)?
   a. void func(int x);
   e. void func(int);

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

   float f;   
   printFun(2.0*f);

   b. void printFun(double)

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;
}
}
```

d. 5

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

e. No value is returned

---

**Problem 2:[5%]**

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

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

1. `77 + c + i + 1L` long int
2. `6.55f + f / 1.5 - 9 / 8` double
3. `'z' - 'z'` int
4. `b + c` int
5. `'t' - 'a' + c` int
6. `77.8f + 4 * 0.5f +45L` float
7. `42L + (int) d + 94.3f + int(4.9)` float
8. `0.0 + f +c` double
9. `5.28L * d * 3 + 4.5` long double
10. `1.5f / d * f + 6.9 * 4L` double
Problem 3:[10%]

// 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

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

#include "sol1_3.h"

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

```cpp
void Material::print(void)
{
    cout << "Material: " << name
        << " Modulus of elasticity = " << modulusElasticity
        << " Poisson ratio = " << ratioPoisson << endl;
}

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

int main()
{
    char name[30] = "Problem 3";
    Material m;

    cout << "Local name : "
        << name;
    cout << " Global name : "
        << ::name;
    cout << " Object name : "
        << m.name;

    cout << "Object: " ;
    m.print();

    cout << "Exiting properly" << endl;
    return EXIT_SUCCESS;
}

Problem 4:[10%]

What the following sets of statements output?

1.

    int x;
    int *y;
    int **z;
```
```cpp
x = 0;
y = &x;
z = &y;

x++;  
(*y)++;
(**z) = 10;
cout << "x is " << x <<  
"*y is " << *y <<  
"**z is " << **z << endl;

x is 10 *y is 10 **z is 10

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;
}

a is 1 b is 1 c is 1  
x is 1 y is 0 z is 1

3.

void swap(int *a, int *b)
{
    int *tmp;
    tmp = a;
```
```c++
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;
}
x[0] = 1  y[0] = 2

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;
}

test1=20  test2=50.8  test3=68

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;
} 

At the beginning
a = 10 b = 20
Before func2
a = 5 b = 40
Problem 5: [40%]

**sol1_5.h**

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

int main();
int getHeight();
void checkHeight(int n);
void drawHourglass(int n);
```

**sol1_5.C**

```c
#include "sol1_5.h"

int main()
{
```
int n;       // Height of figure

n = getHeight();
checkHeight(n);
drawHourglass(n);

}

int getHeight()
{
    int n;
    cout << "Enter height of figure, n : ";
    cin >> n;
    cout << endl;
    return n;
}

void checkHeight(int n)
{
    if (n % 2 == 0 || n < 3)
    {
        cout << "Bad value for n!! exiting..." << endl << endl;
        exit(-1);
    }
}

void drawHourglass(int n)
{
    int i, j;

    for (i=0; i<n; i++) // Print top row of '*'s
        cout << '*';

    cout << endl;

    for ( j = n/2 - 2; j >= 0; j-- ) // Print upper rows
    {
        for ( i = 0; i < n/2-j-1; i++ ) // Move to position of first '*'
            cout << ' ';
        cout << '*'; // Print first '

        cout << ' '; // Print first '

    }
for (i = 0; i < 2*j + 1; i++) // Print spaces
    cout << ' ';
    cout << '*' << endl; // Print second '*' and end line
}
for (i = 0; i < n/2; i++) // Print center row
    cout << ' ';
    cout << '*' << endl;
for (j = 1; j < n/2; j++) // Print lower rows
{
    for (i = 0; i < n/2-j; i++) // Move to position of first '*'
        cout << ' ';
    cout << '*'; // Print first '|' 
    for (i = 0; i < 2*j -1; i++) // Print spaces
        cout << ' ';
    cout << '*' << endl; // Print second '*' and end line
}
for (i=1; i<=n; i++) // Print bottom row of '*'s
    cout << '*';
    cout << endl << endl;
}

Problem 6:[20%]
sol1_6.h

#include <iostream.h> // Problem Set#1 - Problem#6 solution [sol1_6.h]
#include <stdlib.h>

class Complex
private:
  double real;
  double imaginary;

public:
  Complex(double real=0, double imaginary=0)
  {
    cout << "n In Complex(" << real
    << "," << imaginary << ") constructor" << endl;
    this -> real = real ;
    this -> imaginary = imaginary ;
  }

double Complex::get_real(void);
  double Complex::get_imaginary(void);
  void Complex::set_real(double);
  void Complex::set_imaginary(double);
};

sol1_6.C

#include "sol1_6.h" // Problem Set#1 - Problem#6 solution [sol1_6.C]

double Complex::get_real(void)
{
  return real;
}

double Complex::get_imaginary(void)
{
  return imaginary;
}

void Complex::set_real(double real)
{
  this -> real = real ;
}

void Complex::set_imaginary(double im)
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;

    return EXIT_SUCCESS;
}

/******************** Solution output ******************

  In  Complex(0,0) constructor

  c1 = 0 + 0i

  In  Complex(7.25,-8.5) constructor

  c2 = 7.25 + -8.5i

  c1 = 1.7 + -6.7i