

Solved Exercises from exams:

Midterm(1) 1431-1432

1. What is the output of the following program:

```
#include<iostream.h>
void fun(int *b) { *b = *b+10; }
main()
{
    int a=50;
    fun(&a);
    cout<< "a1= "<<a<<endl;
    fun(a);
    cout<< "a2= "<<a<<endl;
    return 0;
}
```

Output: a1= 60

error: type mismatch int , *int

2. a) what is the difference between *(p+1) and *p+1 with declaration:

int *p p pointer to integer
*(p+1) content of the next memory location
*p+1 add 1 to the content of memory location

b) a character array is created as follows:

```
char *cpr = new char[20];
```

How could we delete the memory created using the operator **delete**?

delete [] cpr;

3. 1) What is meant by overloading of a function? When do we use this concept?

Function name overloading: definition of more than one function with the same name. when we call the function, the overloaded function with the correct number of formal parameters and of correct types will be executed.

It is a key concept in object oriented programming.

2) Which two functions are overloaded?

a- void screen(int f); void screen(int);

b- void screen(int); float screen(int);

c- void screen(int); float screen(int , int*);

overloaded

d- float screen(int); float screen(int , int*);

overloaded

4. The following **struct** is declared

```
struct person {
    int date;
    char name[50];
}
```

Write a program in which you define a variable of type person and give it suitable values.

```

int main()
{
    person p1;
    p1.date = 31;
    p1.name = "Alzahrani";
}

```

5. Find errors, if any, in the following statements:

- a) enum Friends{"Tom","Adam","Harry"}
error using " ". Enum Friends {Tom, Adam, Harry}
- b) long float x;
error using long float. double x;
- c) char *cp = vp; // vp is a void pointer
error. Can't assign char pointer to void pointer
int code = three; // three is an enumerator no errors
- d) int *p = new; // allocate memory with new
error using new. int *p = new int;
- e) enum (green,yellow,red);
error. enum colors { green, yellow, red};
- f) int const *p = total;
error. Can't assign pointer to constant to a variable
- g) const int array_size; no errors
- h) int &number = 100;
error. int number = 100;
- i) float *p = new int [10];
error. int *p = new int [10];
- j) char name = "USA";
error char *name = "USA";

6. What is the output of the following program?

```

#include<iostream> using namespace std;
void stat()
{
    int m=0;
    static int n = 0;
    m++;
    n++;
    cout<<" m= "<< m<<" "<<" n = "<< n <<"\n";
}
int main()
{
    stat();
    stat();
    return 0; }

```

Output:

```

m = 1    n = 1
m = 1    n = 2

```

Midterm(2) 1431-1432

Question 1:

Find errors in the following program

```
#include <iostream.h>
class R
{
    int h=1;           value can't be assigned in class declaration
    int w=1;           value can't be assigned in class declaration
public:
    void R(void);       member function with the same name of class
    void a(void) { return h +w; }   return type is int
}
int R::a(void) ;        no prototype is needed

main()
{
    R b, s;
    b.h = 10;           error reference to private data component h
    b.w = 15;           error reference to private data component w
    s.h = s.w = 10;     error reference to private data component h,w
    cout<< b.a() << "\n";
    cout << s.a() << "\n";
    cout << a(s.h, b.w) << "\n";   error call to function(void) in cout
}
```

Question 2:

Rewrite the previous program in “Question 1” that will run properly.

```
#include <iostream.h>
class R
{
    int h;
    int w;
public:
    void set(int num1, int num2) { h = num1;    w = num2; }
    int a(void) { return h+w; }
}
```

```

main()
{
    R b, s;
    b.set(10,15);
    s.set(10,10);
    cout<< b.a() << "\n";
    cout << s.a() << "\n";
}

```

Question 3:

A class rectangle has been declared as:

```
class shape
```

```
{
    float length;
    float width;
};
```

- Declare and define a constructor to initialize the variables of the class.
- Declare and define a method to set the width and height of the shape.
- Declare and define an **in-line** method that finds the area of the **shape** as the product of (length * width).
- In the “**main**” function
- Declare and define an object called “**square**”.
- Display the area of object “**square**”
- How to destroy object “**square**”.

```
class shape
```

```
{
    float length;
    float width;
public:
    shape(void);
    void set(int new_length, int new_width);
    int get_area(void) { return (length * width); } // inline function
    ~shape(void);
};
shape::shape(void)    //Constructor implementation
{
    length = 8;    width = 8;
}
```

```

void shape::set(int new_length, int new_width)
{
    length = new_length;
    width = new_width;
}
shape::~shape(void)    //Destructor
{
    length = 0;
    width = 0;
}

```

```

int main()
{
    shape square;
    cout << "\n The square area is " << square.get_area();
    square.set(10,10);
    cout << "\n The square area is " << square.get_area();
    return 0;
}

```

Question 4

What is the output of the following program:

```

#include<iostream.h>
class box
{
int length;
static int extra_data;
public:
void getdata(int a){length=a; extra_data++;}
void get_extra(void){cout<<"extra_data=";cout<<extra_data<<"\n";}
};
int box::extra_data;
int main( )
{
box a, b, c;
a.get_extra( ); b.get_extra( ); c.get_extra( );
a.getdata(100); a.getdata(200); a.getdata(300);
a.get_extra( ); b.get_extra( ); c.get_extra( );
return 0;
}

```

Output:
extra_data = 0
extra_data = 0
extra_data = 0
extra_data = 3
extra_data = 3
extra_data = 3

Second Semester 1430-1431
Final examination - 151 CSc.

Question 1

Write two programs using functions to swap the values of a pair of integers using:

- **reference variables as arguments**
 - **pointers**
- How can these functions be called.**

- ```
#include<iostream.h>
void swap(int num1, int num2); // prototype
int main()
{
 int x1,x2;
 cout<<"\n enter an integer: ";
 cin>>x;
 cout<<"\n enter a second integer";
 cin>>y;
 if (x1>x2)
 swap (x1,x2);
 return 0; // indicates successful termination
} // end main
void swap(int num1, int num2)
{
 int hold = num1;
 num1 = num2;
 num2 = hold;
} // end function swap
```
- ```
#include<iostream.h>
void swap( int * const, int * const); // prototype
int main()
{
    int x1,x2;
    cout<<"\n enter an integer: ";
    cin>>x;
    cout<<"\n enter a second integer";
    cin>>y;
    if (x1>x2)
        swap (&x1,&x2);
    return 0; // indicates successful termination
} // end main
```

```
void swap( int * const ptr1, int * const ptr2 )
{
    int hold = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = hold;
} // end function swap
```

Question 2

Which of the following statements are illegal?

- a) `int *p = new int(15);` illegal `int *p = new int [15];`
- b) `int*p=new int;`
- c) `int *p = new int[10];`
- d) `delete p[];` illegal `delete [] p;`
- e) `delete p;`
- f) `delete [] p;`

Question 3

A class normal has been declared as:

```
class normal
{
    int x;
    float mean;
    float variance;
public:
}
```

- Declare and define a constructor to initialize variables of the class.
- Declare and define a method that find the pdf of the normal distribution.
- Use these two methods in the “main” function.

```
#include<iostream.h>
#include<math.h>
const float pi = 3.1416;
class normal
{
    int x;
    float mean;
    float variance;
public:
    normal(void);
    float pdf(void);
};
normal::normal(void)      //Constructor implementation
{   x = 165;    mean = 150.0;    variance = 25.0;   }
```

```

float normal::pdf(void)
{   return (1/sqrt(2*pi*variance))*exp(-0.5*pow(x-mean,2)/variance);   }
int main()
{
    normal ex1;
    cout << "\n The pdf value = " << ex1.pdf();
    return 0;
}

```

Question 4

Describe, with examples, the uses of enumeration data types.

Variables with Enumerated user-defined type, can be assigned only one of the values declared in the enumeration.

One example is:

```
enum Game_status { LOSE = 0, WIN, TIE, CANCEL };
```

Another popular enumeration is

```
enum Months { JAN = 1, FEB, MAR, APR, MAY, JUN, JUL, AUG,
SEP, OCT, NOV, DEC };
```

Question 5

The following function prototypes use default arguments. Which of them are illegal?

- a) void prod (int a, int b, int c=10);
- b) void prod(int a=10, int b, int c); **ILLEGAL**
- c) void prod(int a=10, int b=5, int c); **ILLEGAL**
- d) void prod(int a, int b=5, int c=10);
- e) void prod(int a=10, int b=5, int c=10); **ILLEGAL**

Question 6

The following class about complex numbers is declared in the program as:

```

class complex
{
    float x;
    float y;
public:
    complex() { }
    complex(float real; float imag)
        { x=real; y=imag; }
}
int main()
{
    complex C1(2.4,5.0);
    complex C2(1.8, 3.5), C3;
}

```



```
    return 0 ;  
}
```

Add to the program declaration and implementation for functions to show how to:

- **overload binary operator + to add two complex numbers using a friend function.**

```
class complex  
{  
    float x;  
    float y;  
public:  
    complex() { }  
    complex(float real; float imag)  
        { x=real; y=imag; }  
    friend complex operator+(complex z1, complex z2);  
};  
complex operator+(complex z1, complex z2)  
{  
    complex temp;  
    temp.real_part = z1.real_part + z2.real_part;  
    temp.img_part = z1.img_part + z2.img_part;  
    return temp;  
}  
int main()  
{  
    complex C1(2.4,5.0);  
    complex C2(1.8, 3.5), C3;  
    C3 = C1+C2;  
    return 0 ;  
}
```

- **overload binary operator + to add two complex numbers using member function.**

```
class complex  
{  
    float x;  
    float y;  
public:  
    complex() { }  
    complex(float real; float imag)  
        { x=real; y=imag; }  
    complex operator+(complex z);  
};  
complex complex::operator+(complex z)
```

```

{
    complex temp;
    temp.real_part = real_part + z.real_part;
    temp.img_part = img_part + z.img_part;
    return temp;
}
int main()
{
    complex C1(2.4,5.0);
    complex C2(1.8, 3.5), C3;
    C3 = C1.operator+(C2);
    return 0 ;
}

```

Question 7

Find errors , if any, in the following program

```

int main()
{
    int  *pt_int;
    int  pig = 7, dog = 27;
    int  *const ptcon = &pig;
    ptcon = &dog      error constant pointer can't be changed
    ptcon = &pig      error constant pointer
    pig =20;
    char *const name2 = "John";
    *name2 = "John1";
    name2 = "John1";  error constant pointer can't be changed
    char c, c1;        c='x'; c1='y';
    name2 = &c;        error constant pointer can't be changed
    char *const name1=&c;
    name1 = &c1;      error constant pointer can't be changed
    c= 'z';
    return 0;
}

```