

Do not write anything on the question paper except the information required.

Instructions:

1. Read the question carefully, understand the question, and then attempt your answer in the provided answer booklet.
2. Verify that you have Nine (9) printed page of the question paper including this page and Two (2) questions.
3. Calculator sharing is strictly prohibited.
4. Use permanent ink pens only. Any part done using soft pencil will not be marked.
5. Ensure that you do not have any electronic gadget (like mobile phone, smart watch etc.) with you.

Q1: Write the output of the following C++ codes (if the code is correct). If you find any error in the code, please identify, and explain the error/s (Note: do not write output if there is an error). Note that required libraries and main function are already included in the program. [35 marks]

Show dry-run for full credit

i. [5 marks]

```
#include <iostream>
using namespace std;
class A {
public:
    void doSomething() {
        cout << "A" << endl;
    }
};
class B {
public:
    operator A() {
        A a;
        return a;
    }
    void doSomething() {
        cout << "B" << endl;
    }
};
class C {
```

```

public:
    operator B() {
        B b;
        return b;
    }
    operator A() {
        A a;
        return a;
    }
    void doSomething() {
        cout << "C" << endl;
    }
};

void print(void* what, char ch) {
    switch (ch) {
        case 'b':
            ((A)(*((B*)what))).doSomething();
            break;
        case 'c':
            (*((C*)what)).doSomething();
        case 'a':
            ((B)(*((C*)what))).doSomething();
    }
}

int main() {
    C c; B b;
    print((void*)&c, 'c');
    print((void*)&b, 'b');
    print((void*)&c, 'a');
    return 0;
}

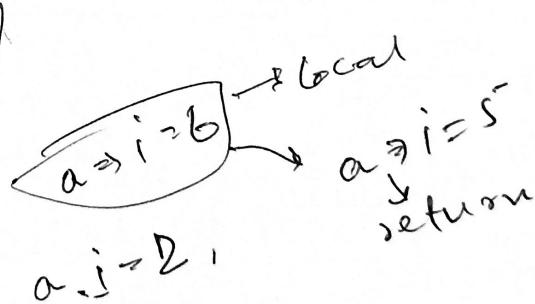
```

ii. [6 marks]

```

#include <iostream>
using namespace std;
class Maze {
private:
    int i;
public:
    - Maze(int i) { // Constructor
        this->i = i;
        cout << " C" << i << "\n";
    }
    - ~Maze() { // Destructor
        cout << " D" << i << "\n";
    }
    - void Y(Maze); // Outline
    - static Maze& X(Maze& b) {
        static Maze a(6); a->i=6;
        b.i++; a.i--;
        return a;
    }
}

```



```

} a = 1;
void Maze::Y(Maze b) {
    Maze c(4);
    b.i++;
    (*this) = b;
}
int main() {
    Maze::X(a).Y(a);
    static Maze a(3);
    a.Y(0);
    return 0;
}
    
```

$\begin{array}{l} \text{c.i = 4} \\ \text{a.i = 6} \\ \cancel{\text{b.i = 2}} \\ \cancel{\text{a.i = 3}} \\ \text{Error} \end{array}$

iii. [5 marks]

```

#include<iostream>
#include<string>
using namespace std;
struct Data {
    int d;
    Data* ptr;
};
class Datachain {
    Data* here;
public:
    Datachain(Data* p = NULL) : here(p) {} // def. const
    void introduce(int x)
    {
        Data* temp = new Data;
        temp->d = x;
        temp->ptr = NULL;
        if (here == NULL)
        {
            here = temp;
        }
        else
        {
            temp->ptr = here;
            here = temp;
        }
        temp = NULL;
    }
    Datachain& show()
    {
        Data* p = here;
        while (p != NULL)
        {
            cout << p->d << " ";
            p = p->ptr;
        }
        cout << endl;
        return *this;
    }
}
    
```

$\begin{array}{l} \text{temp} \rightarrow \text{d = 5} \\ \text{here} \neq \text{Null} \\ \text{temp} \rightarrow \text{d = 1} \\ \text{here} \rightarrow \text{d = 2} \\ | \end{array}$

```

void wonder(Data*& abc, Data*& xyz)
{
    Data* slow = here;
    Data* fast = here->ptr;

    while (fast != NULL) {
        fast = fast->ptr;
        if (fast != NULL) {
            slow = slow->ptr;
            fast = fast->ptr;
        }
    }
    abc = here;
    xyz = slow->ptr;
    slow->ptr = NULL;
}

Datachain& magic()
{
    Data* current = here;
    Data* prev = NULL, * next = NULL;

    while (current != NULL)
    {
        next = current->ptr;
        current->ptr = prev;
        prev = current;
        current = next;
    }
    here = prev;
    return *this;
}

void sooper_magic()
{
    string s = "noitseuq siht fo tuptuo fo dne eht ta yraterceS tpeD fo eman
eht etirw ,skram sunoB roF";
    reverse(s.begin(), s.end());
    cout << s;
}

int main()
{
    Datachain d; here = NULL
    d.introduce(5); d->ptr
    d.introduce(4);
    d.introduce(3);
    d.introduce(2);
    d.introduce(1);
    d.show();
    Data* one;
    Data* two;
    d.wonder(one, two);
    Datachain d2(one);
    Datachain d3(two);
}

```

```
d2.show();
d.magic().show();
d3.show().magic().show().sooper_magic();
}
```

iv. [10 marks]

```
#include<iostream>
using namespace std;
int x = 6;
class A {
    int x;
public:
    A(int x = 0) :x(x + 1) { cout << "A C " << x << endl; }
    ~A() { cout << "~A " << this->x << endl; }
    int& getX() { return this->x; }
}a;
class B {
    A a, * aa;
    int x;
public:
    B() :x(::x), a(--::x), aa(&a) { cout << "Def B C " << x << endl; }
    B(int x) :aa(new A(3)), x(x), a(x++) { cout << "Par B C " << x
<< endl; }
    ~B() { cout << "~B " << B::x << endl; }
}b(x);
class C {
    A& a;
    int x;
public:
    C(A& a) :x(a.getX()++), a(a) { cout << "A C" << endl; }
    ~C() { cout << "~C " << a.getX() << " " << x << endl; }
};
class D {
    C* c;
    B* b;
public:
    D(C* c) :c(c), b(new B) { cout << "C C" << endl; }
    D(B* b, A*&a) :b(new B(*b)), c(new C(*a)) { cout << "D C" <<
endl; }
    ~D() { cout << "~D" << endl; delete b; }
};
int main() {
    D d1(&b, &a);
    D d(new C(a));
    cout << "-----" << endl;
    return 0;
}
```

v. [3+3+3 marks]

Look at the following code:

```
#include <iostream>
using namespace std;
class foo {
private:
    int* ptr;

public:
    foo(int size) {
        ptr = new int[size];
    }

    ~foo() {
        delete[] ptr;
    }

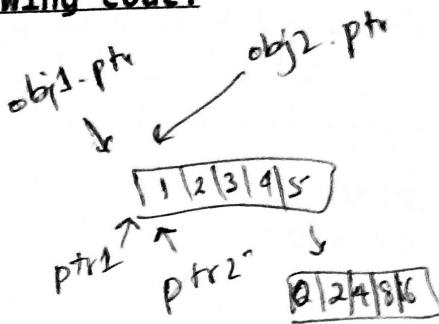
    int* getPtr() const {
        return ptr;
    }
};

int main() {
    foo obj1(5);    obj1.ptr = new int[5]
    foo obj2(obj1); obj2.ptr = new int[5]
    int* ptr1 = obj1.getPtr();    ptr1 =
    for (int i = 0; i < 5; ++i) {
        ptr1[i] = i + 1;
    }

    int* ptr2 = obj2.getPtr();
    for (int i = 0; i < 5; ++i) {
        ptr2[i] = i * 2;
    }

    for (int i = 0; i < 5; ++i) {
        cout << ptr1[i] << " ";
    }
    cout << endl;
}

return 0;
}
```



0-2-4-8-16

Look again. Something changed.

```
#include <iostream>
using namespace std;
```

```

public:
    foo(int size) {
        ptr = new int[size];
    }

    ~foo() {
        delete[] ptr;
    }

    int* getPtr() const {
        return ptr;
    }
};

void globalFunction(foo obj){
    int* ptr = obj.getPtr();
    for (int i = 0; i < 5; ++i) {
        ptr[i] = ptr[i] * -2;
    }
    for (int i = 0; i < 5; ++i) {
        cout << ptr[i] << " ";
    }
    cout << endl;
}

int main() {
    foo obj1(5);

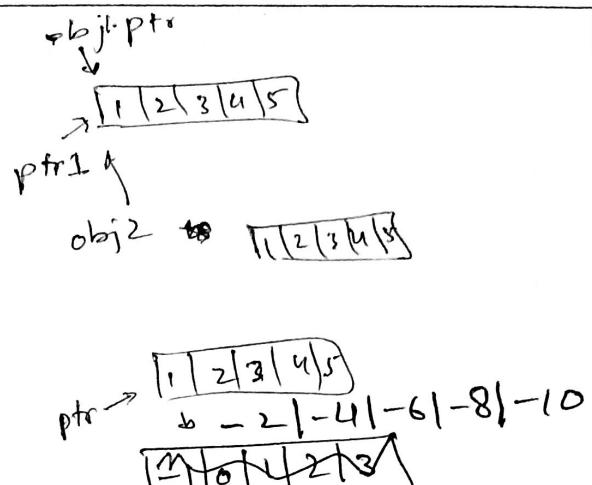
    int* ptr1 = obj1.getPtr();
    for (int i = 0; i < 5; ++i) {
        ptr1[i] = i + 1;
    }

    globalFunction(obj1);

    foo obj2(obj1);
    for (int i = 0; i < 5; ++i) {
        cout << ptr1[i] << " ";
    }
    cout << endl;

    return 0;
}

```



Oops!! Something changed Again

```

#include <iostream>
using namespace std;
class foo {
private:
    int* ptr;
    int size;
}

```

```

public:
    foo(int size) {
        this -> size = size;
        ptr = new int[size];
    }

    ~foo() {
        delete[] ptr;
    }

    foo(const foo& other) {
        this -> size = other.size;
        ptr = new int[this -> size];
        for (int i = 0; i < size; i++)
            ptr[i] = 0;
    }

    int* getPtr() const {
        return ptr;
    }
};

void globalFunction(foo obj){
    int* ptr = obj.getPtr();
    for (int i = 0; i < 5; ++i) {
        ptr[i] = ptr[i] * -2;
    }
    for (int i = 0; i < 5; ++i) {
        cout << ptr[i] << " ";
    }
    cout << endl;
}
    
```

```

int main() {
    foo obj1(5);
    obj1.size = 5;
    obj1.ptr = new int[5];

    int* ptr1 = obj1.getPtr();
    for (int i = 0; i < 5; ++i) {
        ptr1[i] = i + 1;
    }

    globalFunction(obj1);

    foo obj2(obj1);
    for (int i = 0; i < 5; ++i) {
        cout << ptr1[i] << " ";
    }
    cout << endl;

    return 0;
}
    
```

