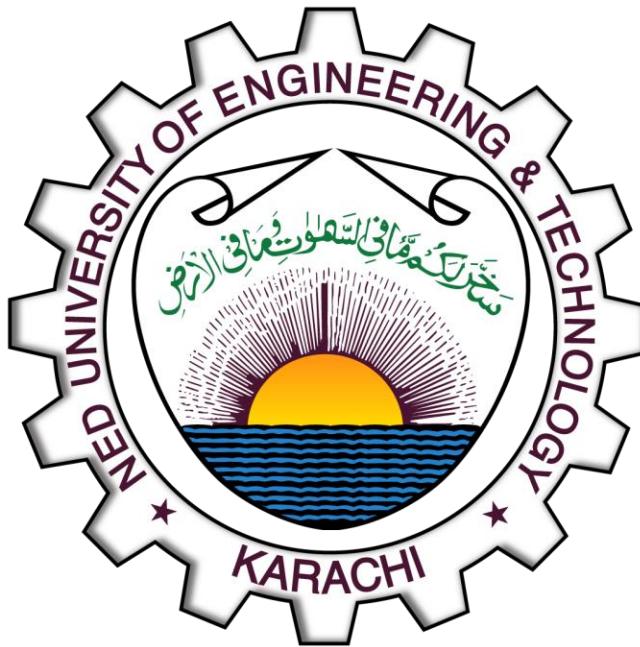


# **OBJECT ORIENTED PROGRAMMING**

**[ICT-260]**



**NAME:** TAHA AHMED MALICK

**ROLL NO:** CT-25183

**DEPARTMENT:** BCIT    **BATCH:** 2025

**YEAR & SECTION:** FSCS-D

**LAB :**3

# **QUESTION#1:**

## **CODE:**

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

class Cmplx
{
public:
    // variables
    float *real, *img;

    // display functions
    void getComplex()
    {
        cout << "\tComplex Number: " << *real << " + " << *img << "i " << endl;
    }

    // constructors
    Cmplx(float a, float b)
    {
        real = new float(a);
        img = new float(b);
    }

    Cmplx(const Cmplx &obj)
    {
        real = new float(*(obj.real));
        img = new float(*(obj.img));
    }
};

int main()
{
    Cmplx num1(2, 5);
    Cmplx num2 = num1;
    cout << "====Copying the Complex Number====" << endl << endl;
    num1.getComplex();
    num2.getComplex();

    *(num1.real) = 8;
    *(num1.img) = 10;
```

```
    cout << endl << "====Changing the FIRST Complex Number====" << endl << endl;
    num1.getComplex();
    num2.getComplex();

    return 0;
}
```

## OUTPUT:

```
====Copying the Complex Number====

    Complex Number: 2 + 5i
    Complex Number: 2 + 5i

====Changing the FIRST Complex Number====

    Complex Number: 8 + 10i
    Complex Number: 2 + 5i
```

## **QUESTION#2:**

### **CODE:**

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

class Character
{
    // member variables
    int *health, *power, *defense;
public:
    // setter methods
    void setHealth(int input)
    {
        *health = input;
    }
    void setPower(int input)
    {
        *power = input;
    }
    void setDefense(int input)
    {
        *defense = input;
    }
    // getter methods
    int getHealth()
    {
        return *health;
    }
    int getPower()
    {
        return *power;
    }
    int getDefense()
    {
        return *defense;
    }
    // Constructors
    Character()
    {
        health = new int(100);
        power = new int(0);
        defense = new int(0);
    }
}
```

```
    }
    Character(int h, int p, int d)
    {
        health = new int(h);
        power = new int(p);
        defense = new int(d);
    }
    Character(const Character &obj)
    {
        health = new int(*(obj.health));
        power = new int(*(obj.power));
        defense = new int(*(obj.defense));
    }
};

int main()
{
    Character a, b(80, 200, 500);
    Character c = a;

    cout << "Character A:" << endl
        << "Health: " << a.getHealth() << endl
        << "Power: " << a.getPower() << endl
        << "Defense: " << a.getDefense() << endl
        << endl;

    cout << "Character B:" << endl
        << "Health: " << b.getHealth() << endl
        << "Power: " << b.getPower() << endl
        << "Defense: " << b.getDefense() << endl
        << endl;

    cout << "Character C:" << endl
        << "Health: " << c.getHealth() << endl
        << "Power: " << c.getPower() << endl
        << "Defense: " << c.getDefense() << endl;

    return 0;
}
```

## **OUTPUT:**

Character A:

Health: 100

Power: 0

Defense: 0

Character B:

Health: 80

Power: 200

Defense: 500

Character C:

Health: 100

Power: 0

Defense: 0

## **QUESTION#3:**

### **CODE:**

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

class TollBooth
{
    int totalCars;
    double totalMoney;
public:

    void payingCar(void)
    {
        totalCars++;
        totalMoney += 0.50;
    }
    void displayTollData()
    {
        cout << "Total Cars: " << totalCars << endl
            << "Total Money: $" << totalMoney << endl
            << endl;
    }

    TollBooth() : totalCars(0), totalMoney(0.0) {}

};

int main()
{
    TollBooth hyderabadStation;

    cout << "Morning Shift:-" << endl;
    for (int i = 0; i < 50; i++)
        hyderabadStation.payingCar();
    hyderabadStation.displayTollData();

    cout << "Evening Shift:-" << endl;
    for (int i = 0; i < 100; i++)
        hyderabadStation.payingCar();
    hyderabadStation.displayTollData();

    cout << "Night Shift:-" << endl;
```

```
    for (int i = 0; i < 10; i++)
        hyderabadStation.payingCar();
    hyderabadStation.displayTollData();
    return 0;
}
```

## **OUTPUT:**

```
Morning Shift:-
Total Cars: 50
Total Money: $25
```

```
Evening Shift:-
Total Cars: 150
Total Money: $75
```

```
Night Shift:-
Total Cars: 160
Total Money: $80
```

## **QUESTION#4:**

### **CODE:**

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

class BookType
{
    string title, author[4], pub;
    int ISBN, noOfCpy, noOfAuth;
    float price;

public:
    // operations on title
    string getTitle()
    {
        return title;
    }
    void setTitle(const string t)
    {
        title = t;
    }
    bool checkTitle(const string t)
    {
        return title == t;
    }
    // operations of number of copies in stock
    int getCopies()
    {
        return noOfCpy;
    }
    void setCopies(const int num)
    {
        noOfCpy = num;
    }
    void updateCopies(const int num)
    {
        noOfCpy += num;
        if (noOfCpy < 0)
            noOfCpy = 0;
    }
    // operations on authors
    void setAuth(const string auth)
```

```
{  
    if (noOfAuth < 4)  
        author[noOfAuth++] = auth;  
    else  
        cout << "Max Limit Reached!!\n";  
}  
void getAuth()  
{  
    cout << "No. of authors: " << noOfAuth << endl;  
    for (int i = 0; i < noOfAuth; i++)  
    {  
        cout << "\tAuthor" << i + 1 << ":" << author[i] << endl;  
    }  
}  
// operations on Publisher  
string getPub()  
{  
    return pub;  
}  
void setPub(const string t)  
{  
    pub = t;  
}  
// operations on book price  
float getPrice()  
{  
    return price;  
}  
void setPrice(float p)  
{  
    price = p;  
}  
// operations on ISBN  
int getISBN()  
{  
    return ISBN;  
}  
void setISBN(int i)  
{  
    ISBN = i;  
}  
// constructor  
BookType()  
{  
    pub = title = "Unknown";
```

```

        for (int i = 0; i < 4; i++)
            author[i] = "N/A";
        ISBN = noOfCpy = noOfAuth = price = 0;
    }
};

void listFeatures(BookType book)
{
    cout << "Title: " << book.getTitle() << endl;
    cout << "Publisher: " << book.getPub() << endl;
    cout << "Price: $" << book.getPrice() << endl;
    cout << "ISBN: " << book.getISBN() << endl;
    book.getAuth();
    int copies = book.getCopies();
    if (copies == 0)
        cout << "\033[1;31mUnavailable.\033[0m" << endl;
    else
        cout << "\033[1;32mIn Stock " << copies << " copies available.\033[0m" <<
endl;
}

int main()
{
    BookType books[100];
    cout << "====Welcome to ABC Book Shop!====" << endl;
    int choice, totalBooks = 0;
    do
    {
        cout << "\nSelect options from below:" << endl;
        cout << "1. Add a book" << endl;
        cout << "2. Display all books" << endl;
        cout << "3. Search by Title" << endl;
        cout << "4. Search by ISBN" << endl;
        cout << "5. Search by Publisher" << endl;
        cout << "6. Update number of copies" << endl;
        cout << "7. Exit" << endl;
        cin >> choice;
        cout << endl;
        switch (choice)
        {
            case 1:
                if (totalBooks < 100)
                {
                    string title, author, pub;
                    int ISBN, noOfCpy, noOfAuth;

```

```

        float price;
        cout << "Enter title of the book: ";
        cin >> title;
        books[totalBooks].setTitle(title);
        cout << "Enter number of authors: ";
        cin >> noOfAuth;
        if (noOfAuth > 4)
            cout << "Only 4 authors allowed at max.\n";
        for (int i = 0; i < noOfAuth && i < 4; i++)
        {
            cout << "Enter the name of author " << i + 1 << ": ";
            cin >> author;
            books[totalBooks].setAuth(author);
        }
        cout << "Enter the name of publisher: ";
        cin >> pub;
        books[totalBooks].setPub(pub);
        cout << "Enter the ISBN number: ";
        cin >> ISBN;
        books[totalBooks].setISBN(ISBN);
        cout << "Enter the number of copies available in stock: ";
        cin >> noOfCpy;
        books[totalBooks].setCopies(noOfCpy);
        cout << "Enter the price of the book: $";
        cin >> price;
        books[totalBooks].setPrice(price);
        totalBooks++;
    }
    else
        cout << "Max limit of books entered." << endl;
    break;
case 2:
    cout << "\033[1;33mTotal number of books (" << totalBooks <<
"):\033[0m" << endl;
    for (int i = 0; i < totalBooks; i++)
    {
        cout << "\033[1;34m===== \033[0m" << endl;
        listFeatures(books[i]);
        cout << "\033[1;34m===== \033[0m" << endl;
    }
    break;
case 3:
{
    string title;
    cout << "Enter the title: ";

```

```

    cin >> title;
    int matches = 0;
    for (int i = 0; i < totalBooks; i++)
        if (books[i].checkTitle(title))
    {
        matches++;
        cout << "\033[1;34m=====\033[0m" <<
endl;
        listFeatures(books[i]);
        cout << "\033[1;34m=====\033[0m" <<
endl;
    }
    cout << "\033[1;33mFound " << matches << " matches.\033[0m" << endl;
}
break;
case 4:
{
    int ISBN;
    cout << "Enter the ISBN: ";
    cin >> ISBN;
    bool found = false;
    for (int i = 0; i < totalBooks; i++)
        if (books[i].getISBN() == ISBN)
    {
        cout << "\033[1;34m=====\033[0m" <<
endl;
        listFeatures(books[i]);
        cout << "\033[1;34m=====\033[0m" <<
endl;
        found = true;
        break;
    }
    if (!found)
        cout << "\033[1;31mThe book with ISBN " << ISBN << " doesn't
exists.\033[0m" << endl;
}
break;
case 5:
{
    string pub;
    cout << "Enter the Publisher: ";
    cin >> pub;
    int matches = 0;
    for (int i = 0; i < totalBooks; i++)
        if (books[i].getPub() == pub)

```

```

    {
        matches++;
        cout << "\033[1;34m===== \033[0m" <<
endl;
        listFeatures(books[i]);
        cout << "\033[1;34m===== \033[0m" <<
endl;
    }
    cout << "\033[1;33mFound " << matches << " matches.\033[0m" << endl;
}
break;
case 6:
{
    int ISBN, i;
    cout << "Enter the ISBN: ";
    cin >> ISBN;
    bool found = false;
    for (i = 0; i < totalBooks; i++)
        if (books[i].getISBN() == ISBN)
    {
        cout << "\033[1;34m===== \033[0m" <<
endl;
        listFeatures(books[i]);
        cout << "\033[1;34m===== \033[0m" <<
endl;
        found = true;
        break;
    }
    if (!found)
        cout << "\033[1;31mThe book with ISBN " << ISBN << " doesn't
exists.\033[0m" << endl;
    else {
        int cpy;
        cout << "Current number of copies: " << books[i].getCopies() <<
endl;
        cout << "Add number of copies: ";
        cin >> cpy;
        books[i].updateCopies(cpy);
    }
}
break;
case 7:
    cout << "Exiting..." << endl;
    break;
default:

```

```
        cout << "\033[1;31mInvalid option\033[0m" << endl;
        break;
    }
} while (choice != 7);
return 0;
}
```

## OUTPUT:

```
====Welcome to ABC Book Shop!=====

Select options from below:
1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit
1

Enter title of the book: abc
Enter number of authors: 2
Enter the name of author 1: def
Enter the name of author 2: ghi
Enter the name of publisher: klm
Enter the ISBN number: 1234
Enter the number of copies available in stock: 24
Enter the price of the book: $9

Select options from below:
1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit
1

Enter title of the book: lmno
Enter number of authors: 5
Only 4 authors allowed at max.
Enter the name of author 1: ijk
Enter the name of author 2: kji
Enter the name of author 3: opq
```

```
Enter the name of author 4: qpo
Enter the name of publisher: ghi
Enter the ISBN number: 5678
Enter the number of copies available in stock: 52
Enter the price of the book: $6
```

Select options from below:

1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit

2

Total number of books (2):

=====

Title: abc

Publisher: klm

Price: \$9

ISBN: 1234

No. of authors: 2

Author1: def

Author2: ghi

In Stock 24 copies available.

=====

=====

Title: lmno

Publisher: ghi

Price: \$6

ISBN: 5678

No. of authors: 4

Author1: ijk

Author2: kji

Author3: opq

Author4: qpo

In Stock 52 copies available.

=====

Select options from below:

1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN

```
5. Search by Publisher  
6. Update number of copies  
7. Exit  
3
```

```
Enter the title: abc  
=====  
Title: abc  
Publisher: klm  
Price: $9  
ISBN: 1234  
No. of authors: 2  
    Author1: def  
    Author2: ghi  
In Stock 24 copies available.  
=====
```

```
Found 1 matches.
```

```
Select options from below:  
1. Add a book  
2. Display all books  
3. Search by Title  
4. Search by ISBN  
5. Search by Publisher  
6. Update number of copies  
7. Exit  
3
```

```
Enter the title: abcd  
Found 0 matches.
```

```
Select options from below:  
1. Add a book  
2. Display all books  
3. Search by Title  
4. Search by ISBN  
5. Search by Publisher  
6. Update number of copies  
7. Exit  
4
```

```
Enter the ISBN: 12345  
The book with ISBN 12345 doesn't exists.
```

```
Select options from below:
```

```
1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit
4
```

```
Enter the ISBN: 1234
=====
Title: abc
Publisher: klm
Price: $9
ISBN: 1234
No. of authors: 2
    Author1: def
    Author2: ghi
In Stock 24 copies available.
=====
```

```
Select options from below:
1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit
5
```

```
Enter the Publisher: klm
=====
Title: abc
Publisher: klm
Price: $9
ISBN: 1234
No. of authors: 2
    Author1: def
    Author2: ghi
In Stock 24 copies available.
=====
Found 1 matches.
```

```
Select options from below:
```

```
1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit
6
```

```
Enter the ISBN: 1234
=====
Title: abc
Publisher: klm
Price: $9
ISBN: 1234
No. of authors: 2
    Author1: def
    Author2: ghi
In Stock 24 copies available.
=====
Current number of copies: 24
Add number of copies: 6
```

```
Select options from below:
1. Add a book
2. Display all books
3. Search by Title
4. Search by ISBN
5. Search by Publisher
6. Update number of copies
7. Exit
2
```

```
Total number of books (2):
=====
Title: abc
Publisher: klm
Price: $9
ISBN: 1234
No. of authors: 2
    Author1: def
    Author2: ghi
In Stock 30 copies available.
=====
=====
```

Title: lmno  
Publisher: ghi  
Price: \$6  
ISBN: 5678  
No. of authors: 4  
    Author1: ijk  
    Author2: kji  
    Author3: opq  
    Author4: qpo  
In Stock 52 copies available.  
=====

Select options from below:

1. Add a book
  2. Display all books
  3. Search by Title
  4. Search by ISBN
  5. Search by Publisher
  6. Update number of copies
  7. Exit
- 8

Invalid option

Select options from below:

1. Add a book
  2. Display all books
  3. Search by Title
  4. Search by ISBN
  5. Search by Publisher
  6. Update number of copies
  7. Exit
- 7

Exiting...