

# **UNIVERSITY INSTITUTE OF COMPUTING**

## **PROJECT REPORT ON AUCTION & BIDDING**

Program Name: BCA

Subject Name/Code: Data Structures(23CAT-201)

Submitted by:Submitted to:

Name: Amit kumar Name: Ms. Preeti

UID: 23BCA10765 Designation: Asst - prof

Section: BCA – 10 “B”

## **ABSTRACT**

### **Introduction:**

- This project is a C++ application that simulates an auction bidding system. It allows users to place bids, stores bid amounts, and identifies the highest bidder.
- Utilizing **priority queues**, it ensures efficient bid organization, making it easy to determine the highest bid at any given time.

## Technique:

The application is developed using C++ with a focus on object-oriented programming (OOP) principles. It employs the following techniques:

1. **To practice object-oriented programming in C++ for real-world application**

**development:**

The project applies object-oriented principles in C++ to build a modular and organized code structure. Using classes for bidders and the auction, this approach improves readability and hands-on experience with OOP concepts in real-world scenarios.

2. **To develop a bidding system that organizes bids efficiently and identifies the highest**

**bidder:**

This project aims to create a streamlined bidding system where users can place bids, and the highest

bid is identified and displayed immediately. The system ensures accurate, real-time organization of bids, enhancing transparency in the auction process.

3. **To utilize priority queues for bid management:**

The priority queue organizes bids by amount, making the highest bid instantly accessible and optimizing performance without extra sorting.

# System Configuration:

- OS: Windows 10 or Linux
- Processor: Intel Core i3 (minimum); Core i5 or higher recommended
- RAM: 4 GB (minimum); 8 GB recommended
- Development Environment: Any C++ IDE (e.g., Visual Studio, Code::Blocks) or Visual Studio Code with a C++ compiler (GCC or Microsoft C++ Compiler)

## SUMMARY

### Input:

#### Main Menu:

- The user is prompted to choose between:
  - 1: Place a bid
  - 2: View highest bidder
  - 3: Exit

#### For Placing Bid:

- Bidder Name: Enter bidder's name.
- Bid Amount: Enter the bid amount.
- After Placing the bid:
  - 1: The system confirms by displaying that the bid has been placed.
  - 2: The bid is added to the priority queue.
  - 0: Exit.
- For Viewing the Highest Bidder:
- The system displays
  - **Highest Bidder's Name:** The name of the bidder with the current highest bid.
  - **Highest Bid Amount:** The amount of the highest bid.

#### Output Example:

- **Place Bid Confirmation:** " Sahil placed a bid of 500 units."
- **Highest Bidder Display:** "Highest Bidder: Sahil with a bid of 500 units.".

#include<iostream>

#include<string>

#include<queue>

using namespace std;

struct bidder{

    string name;

    int bidAmount;

    bidder(string n,int b) : name(n),bidAmount(b){}

    bool operator<(const bidder& other) const{

        return bidAmount <other.bidAmount;

    }

};

class Auction{

private:

    priority\_queue<bidder> bids;

public:

void placebid(string name,int bidAmount){

    bidder newBid(name, bidAmount);

    bids.push(newBid);

    cout<<name<<" place of bid "<<endl;

}

void HighestBidder(){

    if(bids.empty()){

        cout<<"No bids placed"<<endl;

    }else{

```

    bidder Highest = bids.top();

    cout<<"Highest Bidder:"<<Highest.name<<" with a bid of "<<Highest.bidAmount<<endl;

    _}

}

};

int main(){

    Auction auc;

    int numbersofbidders;

    string name;

    int bidamount;

    cout<<"Enter the Numbers of bidder :";

    cin>>numbersofbidders;

    for(int i = 0; i<numbersofbidders;i++){

        cout<<"enter the bidder's name: ";

        cin>>name;

        cout<<"Enter Bid Amount :";

        cin>>bidamount;

        auc.placebid(name.bidamount);

    }

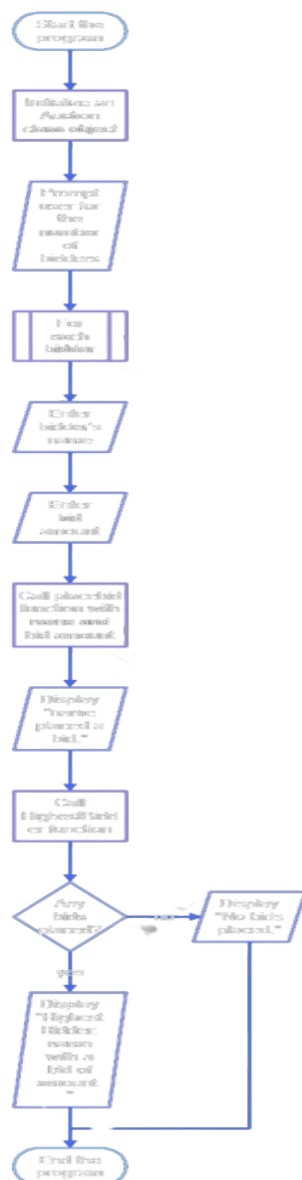
    auc.HighestBidder();

    return 0;

}

```

# Process:



## Output:

Main Menu:

```
Enter the Numbers of bidder :3
```

Bidder Input:

```
Enter the Numbers of bidder :3
enter the bidder's name: sahil
Enter Bid Amount :5000
sahil place of bid
enter the bidder's name: tarmeen
Enter Bid Amount :50001
```

Result & final output:

enter the bidder's name: sahil

Enter Bid Amount :4000

sahil place of bid

enter the bidder's name: tarmeen

Enter Bid Amount :5000

tarmeen place of bid

Highest Bidder:tarmeen with a bid of 5000



# Learning Outcomes for Auction and Bidding System:

## 1. Grasping the Power of Priority Queues:

- Through this project, students get to see how priority queues work in action—managing bids so that the highest one is always on top. This provides a clear understanding of how ordered data structures can be used to solve real-world problems quickly and effectively.

## 1. Applying OOP Concepts in a Practical Way:

- Working with C++ classes and structures in this project, students will strengthen their grasp of object-oriented programming. This experience makes concepts like encapsulation and operator overloading more meaningful and gives them practical tools to build organized, modular code.

## 1. Building Skills with Data Structures:

- By managing bids and user information with data structures like structs and priority\_queue, students gain confidence in dynamically organizing data. This skill becomes invaluable in handling real-time information efficiently, which is key in many real-world applications.

## 1. Enhancing Problem-Solving Abilities:

- Designing an interactive bidding system pushes students to think critically and find effective solutions. Each step, from input management to ensuring the correct bidder information displays, helps in building a problem-solving mindset.

## **1. Understanding Real-World Auction Dynamics:**

- This project also connects students with how real-world auction systems function, helping them apply programming skills to model real-life situations and interactions. It's an exciting way to see programming skills play out in a familiar context.