```
//package Assingment;

public class Lot {

    // class Lot stores all the data on the lots
    // places bids based on the certain criteria

    //Declaring private variables
    private int number;
    private String belongsToAuction;
    private int officialBid = 0;
    private int bidderID = 0;
    private int nextLegalBid = 0;
    private int minIncrementBid = 0;
    private Bid rememberedHigestBid = new Bid(0,0,0);

    // constructor to initialize the variables
    public Lot(int number, String belongsToAuction, int minIncrementBid) {
        this.number = number;
        this.belongsToAuction = belongsToAuction;
        this.minIncrementBid = minIncrementBid;
        this.nextLegalBid = minIncrementBid;
        this.rememberedHigestBid.setLotNumber(number);
    }

    // function to place and process bid on the lot
    public int placeBid(Bid bid){

        if(bid.getAmount() >= this.nextLegalBid){
            //bid was accepted
            return processBid(bid);   // process the bid on the lot based on certain
 rules
        }
        else {
            //Bid was not accepted
            return 1;
        }
    }

    // The main function with logic to process bids
    private int processBid(Bid bid) {

        int status = 1;

        // if the bid amount is higher than the next legal bid then accept bid
        if (bid.getAmount() >= this.nextLegalBid) {

            // The remembered highest bidder is placing the bid on lot
            if(bid.getBidderId() == this.rememberedHigestBid.getBidderId()){

                // check if the bid amount is greater than his previous bid amount o
n the lot
                if(this.rememberedHigestBid.getAmount() < bid.getAmount()){

                    this.rememberedHigestBid.setAmount(bid.getAmount());     //updat
e the current highest remembered amount

                    // validate if there is any chance of poxy bid in the future
                    if(bid.getAmount() >= this.nextLegalBid ){
                        status = 4;
                    }else {
                        status = 3;
                    }
                }

                else {
                    status = 2;   // set status of 2 when we have a bid less than ex
isting bid by the user
                }
            }

            // if the amount is less than equal to the remembered highest bid accept
 the bid
```

```
            else if(bid.getAmount() <= this.rememberedHigestBid.getAmount()){

                // if the new bid is same has the highest remembered bid
                if(bid.getAmount() == this.rememberedHigestBid.getAmount()){

                    // update values for the LOT [auto-bid]
                    this.officialBid  = this.rememberedHigestBid.getAmount();
                    this.nextLegalBid = this.officialBid + this.minIncrementBid;
                    this.bidderID     = this.rememberedHigestBid.getBidderId();

                    // validate if there is any chance of poxy bid in the future
                    status = 2;

                }

                //if the current bid + min increment is less than the remembered hig
hest bid
                if((bid.getAmount()+this.minIncrementBid) <= this.rememberedHigestBi
d.getAmount()){

                    // update values for the LOT
                    this.officialBid  = bid.getAmount()  + this.minIncrementBid;
                    this.nextLegalBid = this.officialBid + this.minIncrementBid;
                    this.bidderID     = this.rememberedHigestBid.getBidderId();

                    // validate if there is any chance of poxy bid in the future
                    status = 2;
                }

                //if the current bid + min increment is less than the remembered hig
hest bid
                if((bid.getAmount()+this.minIncrementBid) > this.rememberedHigestBid
.getAmount()){

                    // update values for the LOT
                    this.officialBid  = bid.getAmount();
                    this.nextLegalBid = this.officialBid + this.minIncrementBid;
                    this.bidderID     = this.rememberedHigestBid.getBidderId();

                    // validate if there is any chance of poxy bid in the future
                    status = 2;
                }

            }


            // if bid amount is higher than the highest remembered bid
            else if(bid.getAmount() > this.rememberedHigestBid.getAmount()){

                // check if this is the first bid on the lot
                if (this.rememberedHigestBid.getAmount() == 0
                && this.rememberedHigestBid.getBidderId() == 0){

                    // update values for the LOT
                    this.officialBid  = nextLegalBid;
                    this.nextLegalBid = this.officialBid + this.minIncrementBid;
                    this.bidderID     = bid.getBidderId();
                    this.rememberedHigestBid.setAmount(bid.getAmount());
                    this.rememberedHigestBid.setBidderId(bid.getBidderId());

                    // validate if there is any chance of poxy bid in the future
                    if(bid.getAmount() >= this.nextLegalBid ){
                        status = 4;
                    }else {
                        status = 3;
                    }
                }

                // check to see if there is room for an automatic bid from the user
end
                else{

                    if(bid.getAmount() < (this.rememberedHigestBid.getAmount()+this.
minIncrementBid)){

                        this.officialBid  = rememberedHigestBid.getAmount();
                        this.nextLegalBid = this.officialBid + minIncrementBid;
                        this.bidderID     = bid.getBidderId();
                        this.rememberedHigestBid.setAmount(bid.getAmount());
                        this.rememberedHigestBid.setBidderId(bid.getBidderId());

                        // validate if there is any chance of poxy bid in the future
                        if(bid.getAmount() >= this.nextLegalBid ){
                            status = 4;
                        }else {
                            status = 3;
                        }

                    }
                    else if(bid.getAmount() >= (this.rememberedHigestBid.getAmount()
+this.minIncrementBid)){

                        this.officialBid  = rememberedHigestBid.getAmount()+minIncre
mentBid;
                        this.nextLegalBid = this.officialBid + minIncrementBid;
                        this.bidderID     = bid.getBidderId();
                        this.rememberedHigestBid.setAmount(bid.getAmount());
                        this.rememberedHigestBid.setBidderId(bid.getBidderId());

                        // validate if there is any chance of poxy bid in the future
                        if(bid.getAmount() >= this.nextLegalBid ){
                            status = 4;
                        }else {
                            status = 3;
                        }


                    }


                }

            }
        }

        return status;

    }


    // Getter and setter functions of the private variables
    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    public String getBelongstoAuction() {
        return belongsToAuction;
    }

    public void setBelongstoAuction(String belongstoAuction) {
        this.belongsToAuction = belongstoAuction;
    }

    public int getOfficialBid() {
        return officialBid;
    }

    public void setOfficialBid(int officialBid) {
        this.officialBid = officialBid;
    }

    public int getBidderID() {
        return bidderID;
    }

    public void setBidderID(int bidderID) {
        this.bidderID = bidderID;
    }

    public Bid setRememberedBids(Bid bid) {
        return rememberedHigestBid;
    }

    public void setRememberedHigestBid(Bid rememberedHigestBid) {
        this.rememberedHigestBid = rememberedHigestBid;
    }

    public Bid getRememberedHigestBid() {
        return rememberedHigestBid;
    }

}
```