Exercise 10-3: make\_bid

using namespace std;

#include <iostream>

#include <iomanip>

#include <string>

class artwork

{

private:

int artworkID = 0;

int ownerID = 0;

string artist = "\_";

string title = "\_";

int minBid = 0;

int currHighBid = 0;

int bidderID = 0;

public:

void auction()

{

cout << "Who is the artist? >> ";

cin.ignore();

getline(cin, artist);

cout << "What is the title? >> ";

getline(cin, title);

cout << "Enter the 4-digit artwork ID: ";

cin >> artworkID;

while (artworkID < 1000 || artworkID > 9999)

{

cout << "The code has to be between 1000 and 9999..." << endl;

cout << "Enter the 4-digit artwork ID: ";

cin >> artworkID;

}

cout << "Enter your 4-digit ID code: ";

cin >> ownerID;

while (ownerID < 1000 || ownerID > 9999)

{

cout << "The code has to be between 1000 and 9999..." << endl;

cout << "Enter the 4-digit owner ID: ";

cin >> ownerID;

}

cout << "What do you want the minimum bid to be? >> $";

cin >> minBid;

}

bool make\_bid(int potentialID, int bid) //parameters: bid & bidderID...

{

if (bid >= minBid && bid > currHighBid)

{

currHighBid = bid;

bidderID = potentialID; //check...

return true; //check...

}

else

{

return false; //check...

}

}

void show(int numEntries)

{

cout << "Bidding Info..." << endl;

cout << "Artist: " << artist << endl;

cout << "Title: " << title << endl;

cout << "Owner ID: " << ownerID << endl;

cout << "Artwork ID: " << artworkID << endl;

cout << "Minimum bid: $" << minBid << endl;

cout << "Current highest bid: $" << currHighBid << endl;

cout << "Highest bidder ID: " << bidderID << endl;

cout << "There are " << numEntries << " more bidding opportunities left " << endl;

cout << "before the auction is over." << endl << endl;

}

void menu()

{

cout << endl;

cout << "==========================================" << endl;

cout << "| What would you like to do? |" << endl;

cout << "==========================================" << endl;

cout << "|Type 1 to enter artwork for the auction |" << endl;

cout << "|Type 2 to bid at the auction |" << endl;

cout << "|Type 3 to view current auction info |" << endl;

cout << "==========================================" << endl;

cout << "Choice: ";

}

};

void main()

{

artwork bidObj;

int bid = 0;

int bidderID;

bool isValid;

int numEntries;

int choice;

cout << "How many bids are going to be logged? >> ";

cin >> numEntries;

do

{

bidObj.menu();

cin >> choice;

cout << endl;

switch (choice)

{

case 1:

bidObj.auction();

break;

case 2:

cout << "Enter your 4-digit bidder ID: ";

cin >> bidderID;

while (bidderID < 1000 || bidderID > 9999)

{

cout << "The code has to be between 1000 and 9999..." << endl;

cout << "Enter the 4-digit bidder ID: ";

cin >> bidderID;

}

cout << "Enter your bid: $";

cin >> bid;

isValid = bidObj.make\_bid(bidderID, bid);

if (isValid)

{

cout << "You are now the highest bidder" << endl;

}

else

{

cout << "You are not the highest bidder" << endl;

}

numEntries--;

break;

case 3:

bidObj.show(numEntries);

break;

default:

cout << "INVALID CHOICE" << endl;

break;

}

} while (numEntries > 0);

system("pause");

}

Exercise 10-4: A Class Called die

**Part A:**

using namespace std;

#include <iostream>

#include <ctime>

#include <string>

class die

{

private:

int value = 1;

public:

void roll()

{

int randNum;

randNum = rand() % 6 + 1;

value = randNum;

}

void show()

{

cout << value << endl;

}

};

void main()

{

die obj;

srand(time(0));

obj.roll();

cout << "You rolled a ";

obj.show();

system("pause");

}

**Part B:**

using namespace std;

#include <iostream>

#include <ctime>

#include <cstdlib>

class die

{

private:

int values[5];

public:

void roll()

{

int randNum;

for (int i = 0; i < 5; i++)

{

randNum = rand() % 6 + 1;

values[i] = randNum;

}

}

void showRoll()

{

for (int i = 0; i < 5; i++)

{

cout << values[i];

}

}

int count(int num)

{

int count;

int roll1 = 0;

int roll2 = 0;

int roll3 = 0;

int roll4 = 0;

int roll5 = 0;

int roll6 = 0;

for (int i = 0; i < 5; i++)

{

if (values[i] == 1)

{

roll1++;

}

if (values[i] == 2)

{

roll2++;

}

if (values[i] == 3)

{

roll3++;

}

if (values[i] == 4)

{

roll4++;

}

if (values[i] == 5)

{

roll5++;

}

if (values[i] == 6)

{

roll6++;

}

}

if (roll6 >= roll5 && roll6 >= roll4 && roll6 >= roll3

&& roll6 >= roll2 && roll6 >= roll1)

{

num = 6;

count = roll6;

}

else if (roll5 > roll6 && roll5 >= roll4 && roll5 >= roll3

&& roll5 >= roll2 && roll5 >= roll1)

{

num = 5;

count = roll5;

}

else if (roll4 > roll6 && roll4 > roll5 && roll4 >= roll3

&& roll4 >= roll2 && roll4 >= roll1)

{

num = 4;

count = roll4;

}

else if (roll3 > roll6 && roll3 > roll5 && roll3 > roll4

&& roll3 >= roll2 && roll3 >= roll1)

{

num = 3;

count = roll3;

}

else if (roll2 > roll6 && roll2 > roll5 && roll2 > roll4

&& roll2 > roll3 && roll2 >= roll1)

{

num = 2;

count = roll2;

}

else if (roll1 > roll6 && roll1 > roll5 && roll1 > roll4

&& roll1 > roll3 && roll1 > roll2)

{

num = 1;

count = roll1;

}

return count;

}

int numVal(int num)

{

int count;

int roll1 = 0;

int roll2 = 0;

int roll3 = 0;

int roll4 = 0;

int roll5 = 0;

int roll6 = 0;

for (int i = 0; i < 5; i++)

{

if (values[i] == 1)

{

roll1++;

}

if (values[i] == 2)

{

roll2++;

}

if (values[i] == 3)

{

roll3++;

}

if (values[i] == 4)

{

roll4++;

}

if (values[i] == 5)

{

roll5++;

}

if (values[i] == 6)

{

roll6++;

}

}

if (roll6 >= roll5 && roll6 >= roll4 && roll6 >= roll3

&& roll6 >= roll2 && roll6 >= roll1)

{

num = 6;

count = roll6;

}

else if (roll5 > roll6 && roll5 >= roll4 && roll5 >= roll3

&& roll5 >= roll2 && roll5 >= roll1)

{

num = 5;

count = roll5;

}

else if (roll4 > roll6 && roll4 > roll5 && roll4 >= roll3

&& roll4 >= roll2 && roll4 >= roll1)

{

num = 4;

count = roll4;

}

else if (roll3 > roll6 && roll3 > roll5 && roll3 > roll4

&& roll3 >= roll2 && roll3 >= roll1)

{

num = 3;

count = roll3;

}

else if (roll2 > roll6 && roll2 > roll5 && roll2 > roll4

&& roll2 > roll3 && roll2 >= roll1)

{

num = 2;

count = roll2;

}

else if (roll1 > roll6 && roll1 > roll5 && roll1 > roll4

&& roll1 > roll3 && roll1 > roll2)

{

num = 1;

count = roll1;

}

return num;

}

void result(int count, int num)

{

cout << "You rolled " << num << " on the die " << count << " time(s)" << endl << endl;

}

};

void main()

{

die obj;

int count = 0;

int num = 0;

char key;

srand(time(0));

do

{

cout << "Roll: ";

obj.roll();

obj.showRoll();

cout << endl;

count = obj.count(num);

num = obj.numVal(num);

obj.result(count, num);

cout << "Keep rolling? Y/N: ";

cin >> key;

} while (key == 'y' || key == 'Y');

system("pause");

}