**LAB ASSIGNMENT**

**WEEK (27April-03 May)**

**Q 1: Define a class in C++ to represent a bank account. Include the**

**following members:**

**Data members:**

**1. Name of the depositor**

**2. Account number**

**3. Type of account**

**4. Balance amount in the account.**

**Member functions:**

**1. To assign initial values**

**2. To deposit an amount**

**3. To withdraw an amount after checking the balance**

**4. To display name and balance**

**Write a main program to test the program.**

***Code:-***

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

class bank

{private:

string name,type,v;

long accountno;

long amt,wamt=0,damt;

public:

void printname()

{

cout<<"Enter the name of depositor"<<endl;

getline (cin,name);

cout<<"Enter the type of account"<<endl;

getline (cin,type);

cout<<"Enter account no"<<endl;

cin>>accountno;

cout<<"Enter the balance in the account"<<endl;

cin>>amt;

cout<<"\n"<<endl;

}

void deposit()

{

cout<<"Enter the amount to deposit"<<endl;

cin>>damt;

amt=amt+damt;

cout<<"INR"<<damt<<" has been deposited in your account."<<endl;

cout<<"\n"<<endl;

}

void withdraw()

{

cout<<"Enter the amount you want to withdraw"<<endl;

cout<<"Current balance is "<<amt<<endl;

cin>>wamt;

amt=amt-wamt;

cout<<"The requested amount has been withdrawn"<<endl;

}

void display()

{

cout<<"\n"<<endl;

cout<<"The name of the account holder and balance in the account is"<<"\n"<<name<<"\n"<<amt<<endl;

}

};

int main()

{

bank obj1;

int i;

obj1.printname();

cout<<"How may I help you?"<<endl;

cout<<"\n1. Deposit the Money"<<endl;

cout<<"2. Withdraw the Money"<<endl;

cout<<"3. Display details of account"<<endl;

cout<<"4.Exit\n"<<endl;

while(1)

{

cout<<"Enter your Choice"<<endl;

cin>>i;

cout<<"\n";

switch (i)

{

case 1:obj1.deposit();

break;

case 2:obj1.withdraw();

break;

case 3:obj1.display();

break;

}

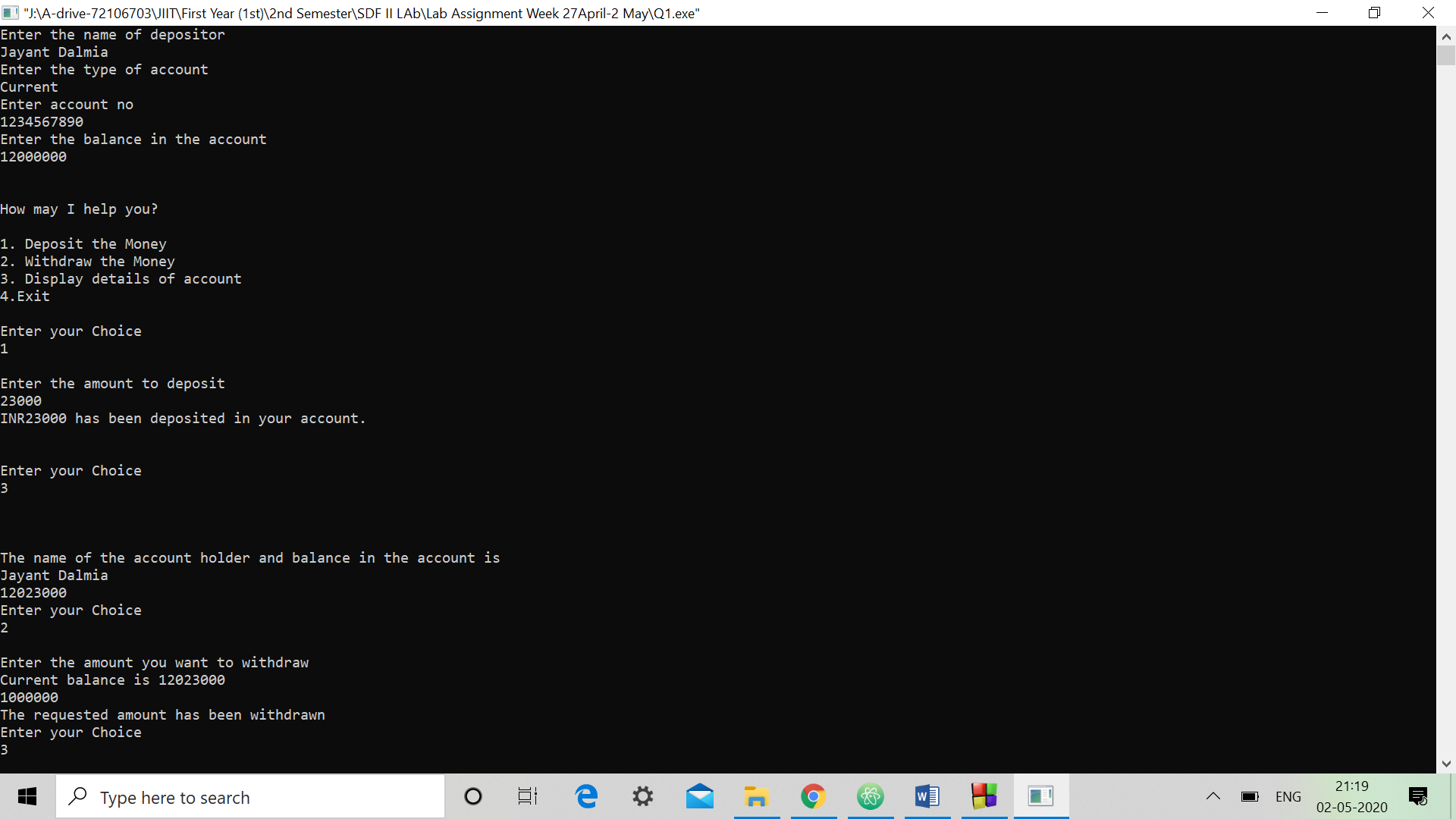
if (i==4)

{break;}

}

return 0;

}



**Q 2: The class Rectangle contains the instance variables height, width,**

**and colour. It implements the following methods:**

**1. Rectangle();**

**2. Rectangle( int size ) ; // Creates a square where height = width = size 3.**

**Rectangle( int height, int width );**

**4. Rectangle( int height, int width, int colour );**

**5. Rectangle( Rectangle otherRectangle );**

**6. void set( Int colour );**

**7. void set( int size );**

**8. void set( int height, int width ) ;**

**9. void set( int height, int width, Colour colour ) ;**

***Code:-***

#include<iostream>

using namespace std;

class Rectangle

{

private:

int height,width,color;

public:

Rectangle();

Rectangle(int);

Rectangle(int,int);

Rectangle(int,int,int);

Rectangle(Rectangle \*r);

void setcol(int);

void setsiz(int);

void sethw(int,int);

void sethwc(int,int,int);

void display();};

Rectangle::Rectangle()

{

sethwc(0,0,0);}

Rectangle::Rectangle(int s)

{

height=s;

width=s;

setcol(0);

}

Rectangle::Rectangle(int h,int w)

{

height=h;

width=w;

setcol(0);}

Rectangle::Rectangle(int h,int w,int c)

{

height=h;

width=w;

color=c;}

Rectangle::Rectangle(Rectangle \*r)

{

height=r->height;

width=r->width;

color=r->color;}

void Rectangle::setcol(int c)

{

color=c;}

void Rectangle::setsiz(int s)

{

height=s;

width=s;}

void Rectangle::sethw(int h,int w)

{

height=h;

width=w;}

void Rectangle::sethwc(int h,int w,int c)

{

height=h;

width=w;

color=c;}

void Rectangle::display()

{

cout<<"Height"<<height<<endl;

cout<<"Width:"<<width<<endl;

cout<<"Color:"<<color<<endl;}

int main()

{

cout<<"Constructors called:-"<<endl;

Rectangle r1,r2(5),r3(5,10),r4(5,10,5),r5(&r3);

cout<<"Object r1 declared as r1"<<endl;

cout<<"Object r1 hold:-"<<endl;

r1.display();

cout<<endl;

cout<<"Object r2 declared as r2(5)"<<endl;

cout<<"Object r2 hold:-"<<endl;

r2.display();

cout<<endl;

cout<<"Object r3 declared as r3(5,10)"<<endl;

cout<<"Object r3 hold:-"<<endl;

r3.display();

cout<<endl;

cout<<"Object r4 declared as r4(5,10,5)"<<endl;

cout<<"Object r4 hold:-"<<endl;

r4.display();

cout<<endl;

cout<<"Object r5 declared as r5(&r3)"<<endl;

cout<<"Object r5 hold:-"<<endl;

r5.display();

r1.setcol(5);

cout<<endl;

cout<<"r1.setcol(5) called"<<endl;

cout<<"Object r1 hold:-"<<endl;

r1.display();

r1.setsiz(8);

cout<<endl;

cout<<"r1.setsiz(8) called"<<endl;

cout<<"Object r1 hold:-"<<endl;

r1.display();

r1.sethw(11,12);

cout<<endl;

cout<<"r1.sethw(11,12) called"<<endl;

cout<<"Object r1 hold:-"<<endl;

r1.display();

r1.sethwc(25,50,10);

cout<<endl;

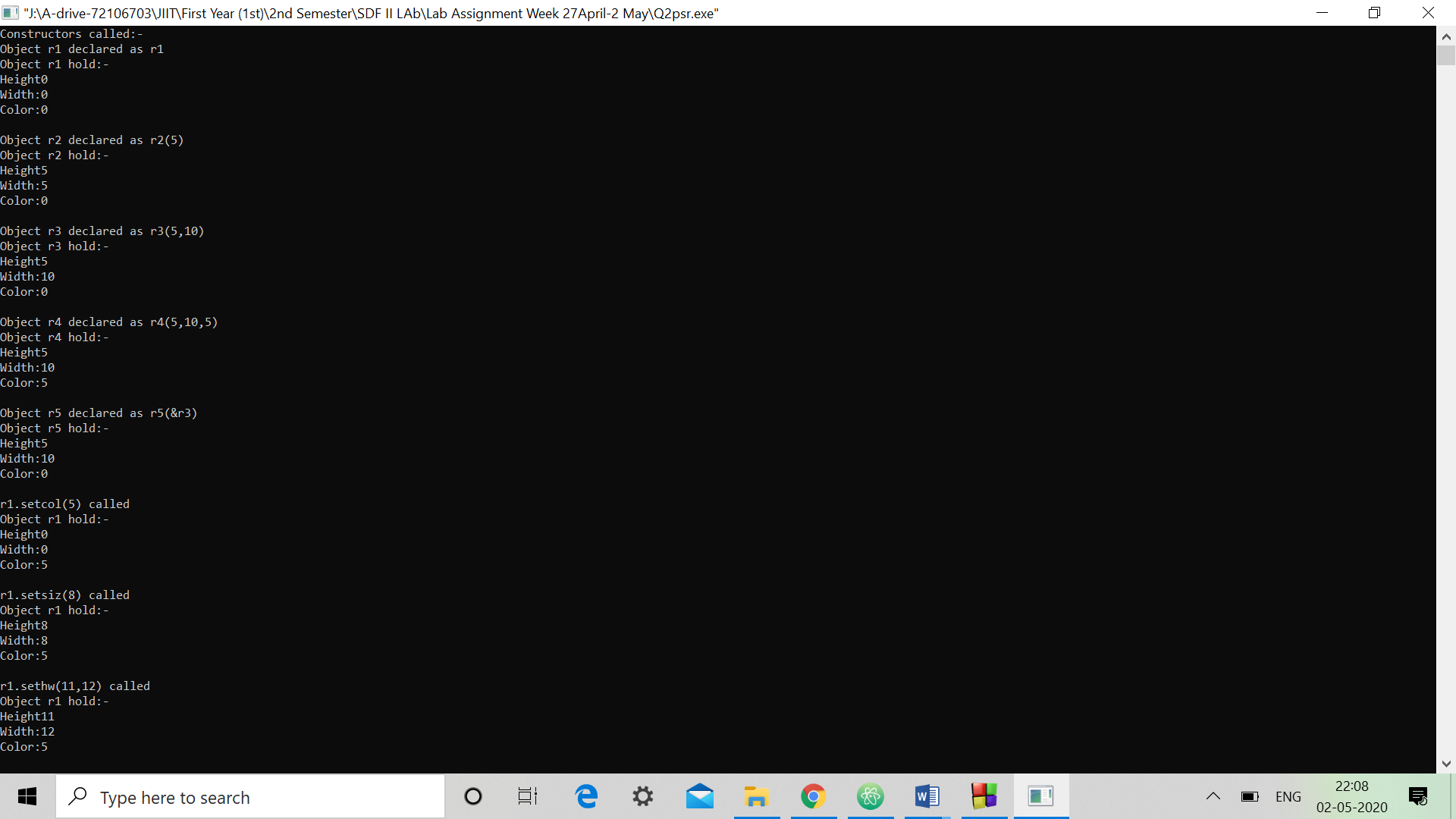
cout<<"r1.sethwc(25,50,10) called"<<endl;

cout<<"Object r1 hold:-"<<endl;

r1.display();

return 0;

}



**Q 3: Design a class called Message. The Message class models a**

**simplified e-mail message, with the following member variables (with the**

**obvious meanings):**

**string from;**

**string to;**

**string text; //the message body**

**int time\_stamp;**

**Give the class the following member functions:**

**1.A constructor that takes sender & recipient, creates an empty**

**message time-stamped with the instant of creation.**

**2.A function that appends a line of text (passed as a string) to**

**the message body.**

***Code:-***

#include<iostream>

#include<time.h>

using namespace std;

class Message

{

private:

string from,to,text,time\_stamp;

public:

Message();

void append(string);

void display();

};

Message::Message()

{

cout<<"Constructor called"<<endl;

cout<<"Sender:";

getline(cin,from);

cout<<"Recipient:";

getline(cin,to);

time\_t t=time(NULL);

time\_stamp=ctime(&t);

text="";

}

void Message::append(string s)

{

text+=s;

}

void Message::display()

{

cout<<"Message Preveiw:-"<<endl;

cout<<"From:"<<from<<endl;

cout<<"To:"<<to<<endl;

cout<<"Message:"<<text<<endl;

cout<<"Time Stamp:"<<time\_stamp<<endl;

}

int main()

{

int ch;

string s;

Message m;

m.display();

cout<<endl<<"Choices:-"<<endl;

cout<<"1.Append Text"<<endl;

cout<<"2.Message Preview"<<endl;

cout<<"3.Exit"<<endl;

while(1)

{

cout<<endl<<"Enter your choice:";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter the text to append:";

fflush(stdin);

getline(cin,s);

m.append(s);

break;

case 2:

m.display();

break;

}

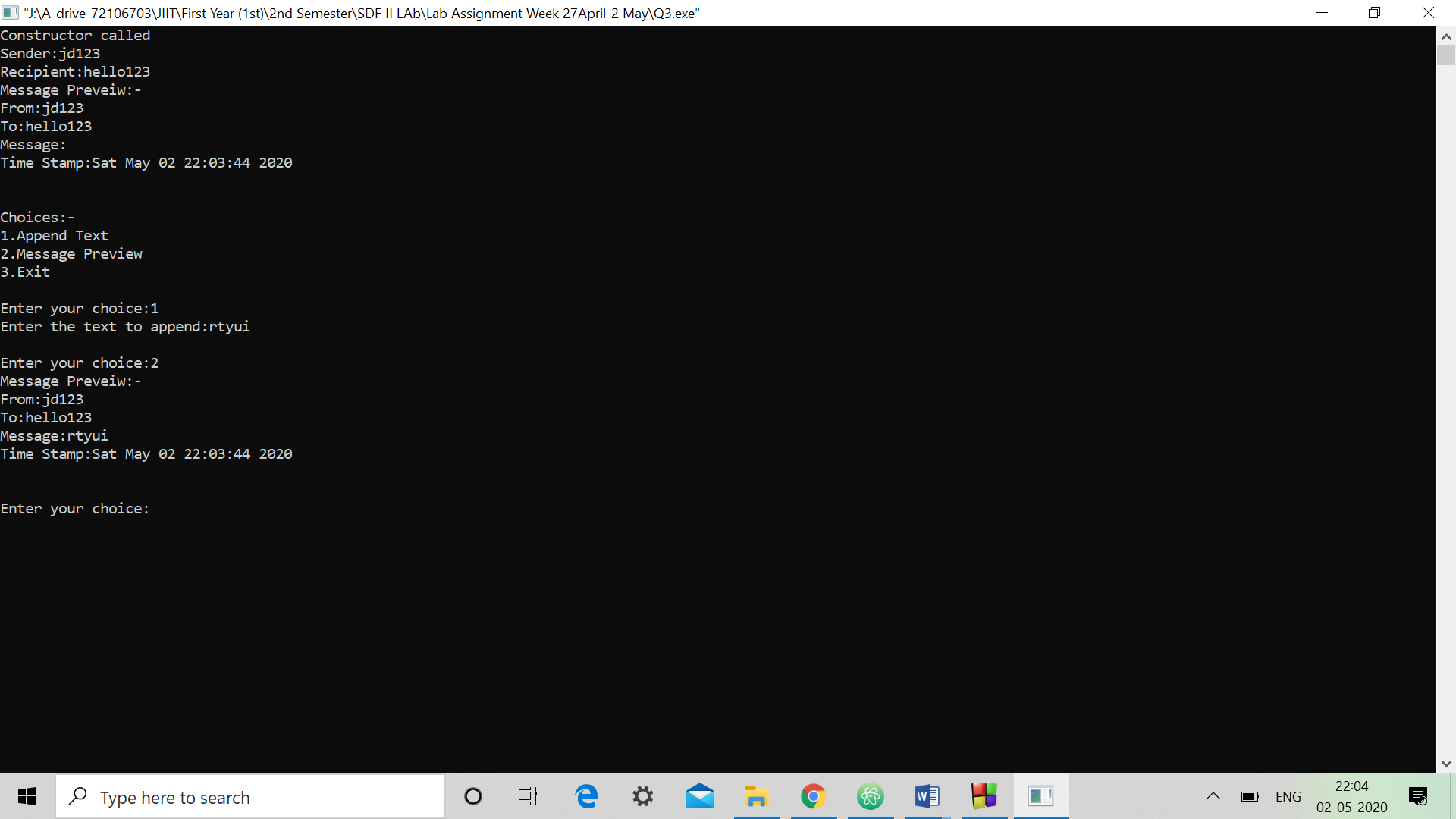
if(ch==3)

break;

}

return 0;

}



**Q 4: Create a class student having name, roll no and semester.**

**1. Create a default constructor**

**2. Create a parameterized constructor with three parameters and**

**default value 1 for semester.**

**3. In main function create an object s1 with name Abc, Roll no.**

**221 and semester 3.**

**4. Now change the semester of s1 from 3 to 4.**

**5. Create an object s2 with values user defined.**

**6. Create a copy constructor for the class**

**7. Create a new object s3 with same values as s1.**

**8. Create an array of n students where n is user defined.**

**9. Write a function check() which is global function and not a**

**member function of class to check whether the data members**

**of s1 and s3 are same or not.**

***Code:-***

#include<iostream>

using namespace std;

class student

{

public:

int sem,roll;

string name;

void setdetails();

void display();

student();

student(int,int,string);

student(student \*);

};

student::student()

{

cout<<"Constructor called"<<endl;

}

student::student(int s,int r,string n)

{

sem=s;

roll=r;

name=n;

}

student::student(student \*s)

{

sem=s->sem;

name=s->name;

roll=s->roll;

}

void student::setdetails()

{

cout<<"Name:";

cin>>name;

cout<<"Roll No:";

cin>>roll;

cout<<"Semester:";

cin>>sem;

}

void student::display()

{

cout<<"Name:"<<name<<endl;

cout<<"Roll No:"<<roll<<endl;

cout<<"Semester:"<<sem<<endl;

}

void check(student \*s1,student \*s2)

{

if(s1->roll==s2->roll && s1->name==s2->name && s1->sem==s2->sem)

cout<<"s1 and s3 are Identical Objects..."<<endl;

else

cout<<"s1 and s3 Non Identical Objects..."<<endl;

}

int main()

{

student s1(3,221,"Abc");

cout<<"s1:-"<<endl;

s1.display();

s1.sem=4;

cout<<"s1:-"<<endl;

s1.display();

student s2;

cout<<"s2:-"<<endl;

s2.setdetails();

cout<<"s2:-"<<endl;

s2.display();

student s3(&s1);

cout<<"s3:-"<<endl;

s3.display();

int i,n;

cout<<"Enter the no of students:";

cin>>n;

student s[n];

for(i=0;i<n;i++)

s[i].setdetails();

check(&s1,&s3);

return 0;

}

