1. Nick Wright CSC285 Problem 5
2. Java Code

/\*The following program will return the result of an expression when passed as a character array.  
\* It will use method parseExpression to calculate the value of the expression, putting the expression through stacks.  
\* The program contains a generic class with pop, push, and peek methods for stacks.  
\* There are 3 classes total, StackParcer3, GenericManagerStacks<T>, and Operatobj.\*/  
  
import java.util.\*;  
import java.lang.\*;  
  
public class StackParcer3 {  
 public static void main(String[] args)throws Exception {  
  
 //ivalue1 contains the values of the integers from the example code  
 int [] ivalue1={7,6,-2,3,1,0,1,2,3,4,5,6,7,8,9};//A=7,B=6,C=-2,D=3,E=1  
 //ivalue2 contains the values of the integers from the assignment  
 int [] ivalue2={8,12,2,3,15,4,0,1,2,3,4,5,6,7,8,9};//A=8,B=12,C=2,D=3,E=15,F=4  
 // A\*3\*(B-C)/((A-C)+D) || this is the original expression that was used in the example code  
 char [] express={'A','\*','3','\*','(','B','-','C',')','/','(','(','A','-','C',')','+','D',')','#'};  
 //express1 = A@(2\*(A-C\*D))+(9\*B/(2\*C+1)-B\*3)+E%(A-F)  
 char[] express1={'A','@','(','2','\*','(','A','-','C','\*','D',')',')','+','(','9','\*','B','/','(','2','\*','C','+','1',')','-','B','\*','3',')','+','E','%','(','A','-','F',')','#'};  
 //express2 = B\*(3@(A-D)%(B-C@D))+4@D\*2  
 char[] express2={'B','\*','(','3','@','(','A','-','D',')','%','(','B','-','C','@','D',')',')','+','4','@','D','\*','2','#'};  
  
 //using method parseExpression to calculate the expressions  
 //parseExpression(express,ivalue1); commenting out the example expression to help reduce the size of the output on the submission document  
 *parseExpression*(express1,ivalue2);  
 *parseExpression*(express2,ivalue2);  
 }//end of main  
  
 /\*created the method parseExpression to parse through the expression character array and return the value.  
 \* this was made an expression so that the while loop did not have to be repeated within the main method every time we wanted to evaluate an expression  
 \* the method will take a character array of an expression and an array of integer values\*/  
 public static void parseExpression(char[] express1, int[] values1){//method to calculate expressions represented as character arrays  
 // now create a stack for the integer operands  
 GenericManagerStacks<Integer> opnd=new GenericManagerStacks<Integer>();  
 // now create a stack for the Operators  
 GenericManagerStacks<Opertobj> oper=new GenericManagerStacks<Opertobj>();  
 System.*out*.println("pushing Operator #with priority -100");  
 Opertobj pnode1=new Opertobj('#',-100); //creating an operator objects and pushing it onto the stack  
 oper.pushnode(pnode1);  
 int oprior,exvalue;//exvalue will hold the value that we want to return after calculation of expression is complete  
 int i, num,ivalu;  
 char [] vart={'A','B','C','D','E','F','0','1','2','3','4','5','6','7', '8','9'}; //the variable array with both letters and numbers  
 char [] opert={'@','%','\*','/','+','-',')','(','#'};//this is the set of operators that will be evaluated in expressions  
 // now create the evaluation priority for the symbols. The higher the priority, the higher the number in the table  
 int [] intvalp={3,2,2,2,1,1,99,-99,-100};//the two operators added (@ and %) have priorities of 3 and 2 respectively.  
 //WE MUST INITIALIZE the OPERATOR STACK so the first Operator can be pushed on. I must put an end of operation on the operator stack.  
 i=0;  
 while(express1[i]!='#') {  
 System.*out*.println("parsing"+express1[i]);  
 if(((express1[i]>='0')&&(express1[i]<='9'))||((express1[i]>='A')&&(express1[i]<='Z')))  
 // Check to see if this character is a variable or an operator.  
 {// we have a variable or a constant  
 System.*out*.println("this is an operand"+express1[i]);  
 // find the character in the vart table that corresponds with the value  
 ivalu=*findval*(express1[i],vart,values1,15);  
 if(ivalu==-99)System.*out*.println("no value in table for"+express1[i]);  
 // now that we have the value we need to place it on the operand stack  
 System.*out*.println("were pushing it on the operand stack"+ivalu);  
 opnd.pushnode(ivalu);  
 }//end of variable stack  
 else {//we are an operator  
 System.*out*.println("this is an operator"+express1[i]);  
 if(express1[i]=='(') {//this is a left parenthesis, push it on the stack  
 // System.out.println("pushing on operator stack"+express[i]);  
 //Create node to push on stack  
 Opertobj pnodeo=new Opertobj(express1[i],-99);  
 oper.pushnode(pnodeo);  
 }else if(express1[i]==')') {// this is a right parenthesis, we must begin to pop operands and pereators  
 //until we find the a left parenthesis (  
 while((oper.peeknode()).operator!='(') {//must pop and evaluate the stuff on operand and operator stack  
 *popevalandpush*(oper,opnd);  
 }  
 // now pop the ( node  
 oper.popnode();  
 }//end of this is a right parenthesis  
 else {//this is not either ( or ) is is another operator  
 oprior=*findval*(express1[i],opert,intvalp,7);  
 System.*out*.println("peeking at top of stack"+(oper.peeknode()).priority);  
 //\*\*\*\*\*\*\*\*\*\*oprior MUST BE STRICTLY GREATER THAN BEFORE WE CAN PUT IT ON THE STACK\*\*\*\*\*\*\*\*  
 while(oprior<=(oper.peeknode()).priority)*popevalandpush*(oper,opnd);  
 //now push this operator on the stack.  
 System.*out*.println("pushing Operator"+express1[i]+"with priority"+oprior);  
 Opertobj pnodeo=new Opertobj(express1[i],oprior);  
 oper.pushnode(pnodeo);  
 }//this is the end of this is not () operator  
 }//end of on operator stack  
 i++;  
 }//end of while express loop  
 //we have found the # in the evaluation now we must evaluate the operator stack  
 while((oper.peeknode()).operator!='#') {//we are finishing up operator stack  
 *popevalandpush*(oper,opnd);  
 }// end of finishing up operator stack  
 //we're done, get value of opnd stack and print  
 exvalue=opnd.popnode();  
 System.*out*.println ("the value for this expression is "+exvalue+"\n");//prints out the expression value  
 }  
  
 public static int IntEval(int oper1, char oper, int oper2) {//This is an evaluator for binary operators operating on integers.  
 int result;  
 //switch statement is used to calculate the value between two integers with an operator  
 switch(oper) {  
 case '+': result=oper1+oper2;  
 System.*out*.println("\*\*\*eval"+oper1+oper+oper2+"\*result\*"+result);  
 return result;  
  
 case '-': result=oper1-oper2;  
 System.*out*.println("\*\*\*eval"+oper1+oper+oper2+"\*result\*"+result);  
 return result;  
  
 case '\*': result=oper1\*oper2;  
 System.*out*.println("\*\*\*eval"+oper1+oper+oper2+"\*result\*"+result);  
 return result;  
  
 case '/': if (oper2!=0) {result=oper1/oper2;  
 System.*out*.println("\*\*\*eval"+oper1+oper+oper2+"\*result\*"+result);  
 return result;  
 } else {// we have an attemped divide by zero  
 System.*out*.println("attempted divide by zero not allowed");  
 return -99;  
 }  
  
 case '@': result =(int)Math.*pow*(oper1,oper2);  
 System.*out*.println("\*\*\*eval"+oper1+oper+oper2+"\*result\*"+result);  
 return result;  
  
 case '%': result = oper1%oper2;  
 System.*out*.println("\*\*\*eval"+oper1+oper+oper2+"\*result\*"+result);  
 return result;  
  
 default: System.*out*.println("bad operator"+oper);  
 return -99;  
 }//end of switch(oper)  
 }//end of IntEval  
  
 public static int findval(char x, char [] vtab, int [] valtb, int last) {  
 int i, vreturn=-99;  
 // this finds the character x in the value table vtab and returns the  
 //correspond interger value table from valtb  
 for(i=0;i<=last; i++)  
 if(vtab[i]==x)vreturn=valtb[i];  
 System.*out*.println("found this char"+x+"value is"+vreturn);  
 return vreturn;  
 }//end of findval;  
  
 public static void popevalandpush(GenericManagerStacks<Opertobj> x, GenericManagerStacks<Integer> y) {//this is the start of pop and push  
 int a,b,c;  
 char operandx;  
 operandx=(x.popnode()).Getopert();  
 a=y.popnode();  
 b=y.popnode();  
 System.*out*.println("in popeval"+b+operandx+a);  
 c=*IntEval*(b,operandx,a);  
 //now push the value back on the stack for integers  
 y.pushnode(c);  
 return;  
 }//This is the end of popevalandpush  
}// this is the end of Stackparcer3 class  
  
class GenericManagerStacks<T>{//generic manager class for stacks  
 protected ArrayList<T> mystack;  
 protected int number;  
  
 public GenericManagerStacks() {// this is the generic constructor  
 number=0;//mcount is the next available value in array myarray  
 mystack=new ArrayList<T>(100);//creates an initial arraylist of 100  
 }  
  
 public int getnumber(){return number;}  
 public int pushnode(T x) {  
 System.*out*.println("in pushnode"+number+"x is"+x);  
 //this pushes a node on the stack. It will always add to the front(top) of the stack  
 mystack.add(number,x);  
 number++;  
 System.*out*.println("leaving pushnode");  
 return number;  
 }// end of pushnode  
  
 public T popnode() {//this function returns the first node in the list  
 T nodeval;//this is the value in the node to be popped  
 // find the node at the head of the list  
 nodeval=mystack.get(number-1);  
 //now pop the node by taking it off the list and moving head  
 mystack.remove(number-1);  
 number--;  
 //now return the value of this node.  
 return nodeval;  
 }// this is the end of popnode  
  
 public T peeknode() {//this function returns the contents of the top of the stack. It does not  
 //pop the node, just allows the user to look (peek) at the contents of the  
 //first node on the stack.  
 T nodeval; //this is the value to be peeked  
 nodeval=mystack.get(number-1);  
 return nodeval;  
 }//this is the end of peeknode  
  
 boolean stackempty(){if(number==0)return true;  
 else return false;}  
}//end of GenericManager class  
  
class Opertobj {// this is an operator class it will hold a character operator and it's stack priority  
 protected char operator;  
 protected int priority;  
  
 public Opertobj(char opert, int pri) {//this is the constructor for the operator object  
 operator=opert;  
 priority=pri;  
 }  
 public int Getprior(){return priority;};  
 public char Getopert(){return operator;};  
}//this is the end of the operator class

1. Input Files

No Input Files used

1. Output Files

pushing Operator #with priority -100

in pushnode0x isOpertobj@70177ecd

leaving pushnode

parsingA

this is an operandA

found this charAvalue is8

were pushing it on the operand stack8

in pushnode0x is8

leaving pushnode

parsing@

this is an operator@

found this char@value is3

peeking at top of stack-100

pushing Operator@with priority3

in pushnode1x isOpertobj@179d3b25

leaving pushnode

parsing(

this is an operator(

in pushnode2x isOpertobj@254989ff

leaving pushnode

parsing2

this is an operand2

found this char2value is2

were pushing it on the operand stack2

in pushnode1x is2

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack-99

pushing Operator\*with priority2

in pushnode3x isOpertobj@5d099f62

leaving pushnode

parsing(

this is an operator(

in pushnode4x isOpertobj@37f8bb67

leaving pushnode

parsingA

this is an operandA

found this charAvalue is8

were pushing it on the operand stack8

in pushnode2x is8

leaving pushnode

parsing-

this is an operator-

found this char-value is1

peeking at top of stack-99

pushing Operator-with priority1

in pushnode5x isOpertobj@49c2faae

leaving pushnode

parsingC

this is an operandC

found this charCvalue is2

were pushing it on the operand stack2

in pushnode3x is2

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack1

pushing Operator\*with priority2

in pushnode6x isOpertobj@20ad9418

leaving pushnode

parsingD

this is an operandD

found this charDvalue is3

were pushing it on the operand stack3

in pushnode4x is3

leaving pushnode

parsing)

this is an operator)

in popeval2\*3

\*\*\*eval2\*3\*result\*6

in pushnode3x is6

leaving pushnode

in popeval8-6

\*\*\*eval8-6\*result\*2

in pushnode2x is2

leaving pushnode

parsing)

this is an operator)

in popeval2\*2

\*\*\*eval2\*2\*result\*4

in pushnode1x is4

leaving pushnode

parsing+

this is an operator+

found this char+value is1

peeking at top of stack3

in popeval8@4

\*\*\*eval8@4\*result\*4096

in pushnode0x is4096

leaving pushnode

pushing Operator+with priority1

in pushnode1x isOpertobj@4459eb14

leaving pushnode

parsing(

this is an operator(

in pushnode2x isOpertobj@5a2e4553

leaving pushnode

parsing9

this is an operand9

found this char9value is9

were pushing it on the operand stack9

in pushnode1x is9

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack-99

pushing Operator\*with priority2

in pushnode3x isOpertobj@28c97a5

leaving pushnode

parsingB

this is an operandB

found this charBvalue is12

were pushing it on the operand stack12

in pushnode2x is12

leaving pushnode

parsing/

this is an operator/

found this char/value is2

peeking at top of stack2

in popeval9\*12

\*\*\*eval9\*12\*result\*108

in pushnode1x is108

leaving pushnode

pushing Operator/with priority2

in pushnode3x isOpertobj@6659c656

leaving pushnode

parsing(

this is an operator(

in pushnode4x isOpertobj@6d5380c2

leaving pushnode

parsing2

this is an operand2

found this char2value is2

were pushing it on the operand stack2

in pushnode2x is2

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack-99

pushing Operator\*with priority2

in pushnode5x isOpertobj@45ff54e6

leaving pushnode

parsingC

this is an operandC

found this charCvalue is2

were pushing it on the operand stack2

in pushnode3x is2

leaving pushnode

parsing+

this is an operator+

found this char+value is1

peeking at top of stack2

in popeval2\*2

\*\*\*eval2\*2\*result\*4

in pushnode2x is4

leaving pushnode

pushing Operator+with priority1

in pushnode5x isOpertobj@2328c243

leaving pushnode

parsing1

this is an operand1

found this char1value is1

were pushing it on the operand stack1

in pushnode3x is1

leaving pushnode

parsing)

this is an operator)

in popeval4+1

\*\*\*eval4+1\*result\*5

in pushnode2x is5

leaving pushnode

parsing-

this is an operator-

found this char-value is1

peeking at top of stack2

in popeval108/5

\*\*\*eval108/5\*result\*21

in pushnode1x is21

leaving pushnode

pushing Operator-with priority1

in pushnode3x isOpertobj@bebdb06

leaving pushnode

parsingB

this is an operandB

found this charBvalue is12

were pushing it on the operand stack12

in pushnode2x is12

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack1

pushing Operator\*with priority2

in pushnode4x isOpertobj@7a4f0f29

leaving pushnode

parsing3

this is an operand3

found this char3value is3

were pushing it on the operand stack3

in pushnode3x is3

leaving pushnode

parsing)

this is an operator)

in popeval12\*3

\*\*\*eval12\*3\*result\*36

in pushnode2x is36

leaving pushnode

in popeval21-36

\*\*\*eval21-36\*result\*-15

in pushnode1x is-15

leaving pushnode

parsing+

this is an operator+

found this char+value is1

peeking at top of stack1

in popeval4096+-15

\*\*\*eval4096+-15\*result\*4081

in pushnode0x is4081

leaving pushnode

pushing Operator+with priority1

in pushnode1x isOpertobj@45283ce2

leaving pushnode

parsingE

this is an operandE

found this charEvalue is15

were pushing it on the operand stack15

in pushnode1x is15

leaving pushnode

parsing%

this is an operator%

found this char%value is2

peeking at top of stack1

pushing Operator%with priority2

in pushnode2x isOpertobj@2077d4de

leaving pushnode

parsing(

this is an operator(

in pushnode3x isOpertobj@7591083d

leaving pushnode

parsingA

this is an operandA

found this charAvalue is8

were pushing it on the operand stack8

in pushnode2x is8

leaving pushnode

parsing-

this is an operator-

found this char-value is1

peeking at top of stack-99

pushing Operator-with priority1

in pushnode4x isOpertobj@77a567e1

leaving pushnode

parsingF

this is an operandF

found this charFvalue is4

were pushing it on the operand stack4

in pushnode3x is4

leaving pushnode

parsing)

this is an operator)

in popeval8-4

\*\*\*eval8-4\*result\*4

in pushnode2x is4

leaving pushnode

in popeval15%4

\*\*\*eval15%4\*result\*3

in pushnode1x is3

leaving pushnode

in popeval4081+3

\*\*\*eval4081+3\*result\*4084

in pushnode0x is4084

leaving pushnode

the value for this expression is 4084

pushing Operator #with priority -100

in pushnode0x isOpertobj@2d8e6db6

leaving pushnode

parsingB

this is an operandB

found this charBvalue is12

were pushing it on the operand stack12

in pushnode0x is12

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack-100

pushing Operator\*with priority2

in pushnode1x isOpertobj@23ab930d

leaving pushnode

parsing(

this is an operator(

in pushnode2x isOpertobj@4534b60d

leaving pushnode

parsing3

this is an operand3

found this char3value is3

were pushing it on the operand stack3

in pushnode1x is3

leaving pushnode

parsing@

this is an operator@

found this char@value is3

peeking at top of stack-99

pushing Operator@with priority3

in pushnode3x isOpertobj@3fa77460

leaving pushnode

parsing(

this is an operator(

in pushnode4x isOpertobj@619a5dff

leaving pushnode

parsingA

this is an operandA

found this charAvalue is8

were pushing it on the operand stack8

in pushnode2x is8

leaving pushnode

parsing-

this is an operator-

found this char-value is1

peeking at top of stack-99

pushing Operator-with priority1

in pushnode5x isOpertobj@1ed6993a

leaving pushnode

parsingD

this is an operandD

found this charDvalue is3

were pushing it on the operand stack3

in pushnode3x is3

leaving pushnode

parsing)

this is an operator)

in popeval8-3

\*\*\*eval8-3\*result\*5

in pushnode2x is5

leaving pushnode

parsing%

this is an operator%

found this char%value is2

peeking at top of stack3

in popeval3@5

\*\*\*eval3@5\*result\*243

in pushnode1x is243

leaving pushnode

pushing Operator%with priority2

in pushnode3x isOpertobj@7e32c033

leaving pushnode

parsing(

this is an operator(

in pushnode4x isOpertobj@7ab2bfe1

leaving pushnode

parsingB

this is an operandB

found this charBvalue is12

were pushing it on the operand stack12

in pushnode2x is12

leaving pushnode

parsing-

this is an operator-

found this char-value is1

peeking at top of stack-99

pushing Operator-with priority1

in pushnode5x isOpertobj@497470ed

leaving pushnode

parsingC

this is an operandC

found this charCvalue is2

were pushing it on the operand stack2

in pushnode3x is2

leaving pushnode

parsing@

this is an operator@

found this char@value is3

peeking at top of stack1

pushing Operator@with priority3

in pushnode6x isOpertobj@63c12fb0

leaving pushnode

parsingD

this is an operandD

found this charDvalue is3

were pushing it on the operand stack3

in pushnode4x is3

leaving pushnode

parsing)

this is an operator)

in popeval2@3

\*\*\*eval2@3\*result\*8

in pushnode3x is8

leaving pushnode

in popeval12-8

\*\*\*eval12-8\*result\*4

in pushnode2x is4

leaving pushnode

parsing)

this is an operator)

in popeval243%4

\*\*\*eval243%4\*result\*3

in pushnode1x is3

leaving pushnode

parsing+

this is an operator+

found this char+value is1

peeking at top of stack2

in popeval12\*3

\*\*\*eval12\*3\*result\*36

in pushnode0x is36

leaving pushnode

pushing Operator+with priority1

in pushnode1x isOpertobj@b1a58a3

leaving pushnode

parsing4

this is an operand4

found this char4value is4

were pushing it on the operand stack4

in pushnode1x is4

leaving pushnode

parsing@

this is an operator@

found this char@value is3

peeking at top of stack1

pushing Operator@with priority3

in pushnode2x isOpertobj@6438a396

leaving pushnode

parsingD

this is an operandD

found this charDvalue is3

were pushing it on the operand stack3

in pushnode2x is3

leaving pushnode

parsing\*

this is an operator\*

found this char\*value is2

peeking at top of stack3

in popeval4@3

\*\*\*eval4@3\*result\*64

in pushnode1x is64

leaving pushnode

pushing Operator\*with priority2

in pushnode2x isOpertobj@e2144e4

leaving pushnode

parsing2

this is an operand2

found this char2value is2

were pushing it on the operand stack2

in pushnode2x is2

leaving pushnode

in popeval64\*2

\*\*\*eval64\*2\*result\*128

in pushnode1x is128

leaving pushnode

in popeval36+128

\*\*\*eval36+128\*result\*164

in pushnode0x is164

leaving pushnode

the value for this expression is 164

Process finished with exit code 0

1. Class Documentation

public **class StackParcer3**

**StackParcer3 Data**

The following data are located withing the main method:

int[] ivalue1={7,6,-2,3,1,0,1,2,3,4,5,6,7,8,9};

int[] ivalue2={8,12,2,3,15,4,0,1,2,3,4,5,6,7,8,9};

char[] express={'A','\*','3','\*','(','B','-','C',')','/','(','(','A','-','C',')','+','D',')','#'};

char[] express1={'A','@','(','2','\*','(','A','-','C','\*','D',')',')','+','(','9','\*','B','/','(','2','\*','C','+','1',')','-','B','\*','3',')','+','E','%','(','A','-','F',')','#'};

char[] express2={'B','\*','(','3','@','(','A','-','D',')','%','(','B','-','C','@','D',')',')','+','4','@','D','\*','2','#'};

**StackParcer3 Functions**

public static void main(String[] args) throws exception (main method)

public static void parseExpression(char[] express1, int[] values1)

public static int IntEval(int oper1, char oper, int oper2)

public static int findval(char x, char[] vtab, int[] valtb, int last)

Public static void popevalandpush(GenericManagerStacks<Operatobj> x, GenericManagerStacks<Integer> y)

**class GenericManagerStacks**<T>

**GenericManagerStacks Data**

protected ArrayList<T> mystack;

protected int number;

**GenericManagerStacks Functions**

public GenericManagerStacks()

public int getnumber()

public int pushnode(T x)

public T popnode

public T peeknode

boolean stackempty()

**class Operatobj**

**Operatobj Data**

protected char operator;

protected int priority

**Operatobj Functions**

public Operatobj(char opert, int pri)

public int Getprior()

public char Getopert()