IS 1001 – Programming and Problem Solving – Tutorial 7

Stack and Queue Answers

```
##Stack class that can be used to all questions
class Stack:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop()

    def peek(self):
        return self.items[len(self.items)-1]

    def size(self):
        return len(self.items)

    def printStack(self):
        print(self.items)
```

Question 1

```
##Question 1

##part 1

s=Stack()


##part 2

s.push("A")

s.push("B")

s.push("C")

s.printStack()

##part 3

print(s.size())

##part 4

print(s.peek())

##part 5

s.pop()

s.printStack()

##part 6

print(s.isEmpty())

##part 7

s.push("D")

s.push("E")

s.printStack()

print(s.size())
```

Question 2

```
##Question 2
s=Stack()
x=input("Do you want to enter a website link  ")
while (x=="Yes"):
   y=input()
   s.push(y)
   s.printStack()
   x=input("Do you want Browse another web site")
print("Your last ten visited websites are as follows : ")
z=10
while(z>=1):

   if(s.isEmpty()!=True):
      print(s.pop())
   z=z-1
```

Question 3

```
##Question 3
StudentList=['25','22','20','18','15']
NewStudentList=[]
s=Stack()
x=""
z=0
for i in StudentList:
    s.push(i)
```

```
while(z<=len(StudentList)):
    if(s.isEmpty()!=True):
        x=s.pop()
        NewStudentList.append(x)

    z=z+1
print("NewStudentList",NewStudentList)
```

Question 4

```
##Question 4


def parChecker(symbolString):
    s = Stack()
    balanced = True
    index = 0
    while index < len(symbolString) and balanced:
        symbol = symbolString[index]
        if symbol == '{' or symbol == '(' or symbol == '[':
            s.push(symbol)
        elif symbol == '}' or symbol == ')' or symbol == ']':
            if s.isEmpty():
                balanced = False
            else:
                last = s.peek()
                if(symbol == '}' and last == '{' or symbol == ')' and last == '(' or symbol == ']' and last == '['):
                    s.pop()

        index = index + 1
```

```
    if balanced and s.isEmpty():
        return True
    else:
        return False

print(parChecker('((()))'))
print(parChecker('(()'))
print(parChecker('{(20+15)/(4+1)}'))
print(parChecker('{(20+15)/(4+1}'))
```