



CS213

Object Oriented Programming

Big Real

2023


Dr Mohammed EL-Ramly
m.elramly@fci-cu.edu.eg



Team Members :

Name	ID
Osama Refaat Sayed	20221015
Mostafa Mohamed Anwar	20221153
Ziad Tawfik Abdelnabi	20221066





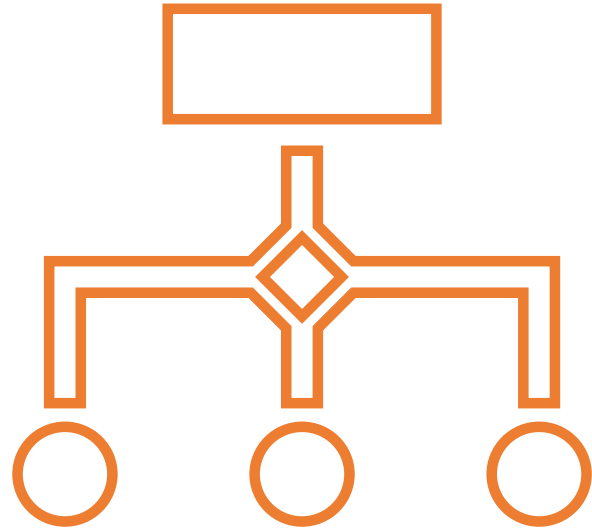
1 2 3 4 5
6 7 8 9 0

Big Real Task

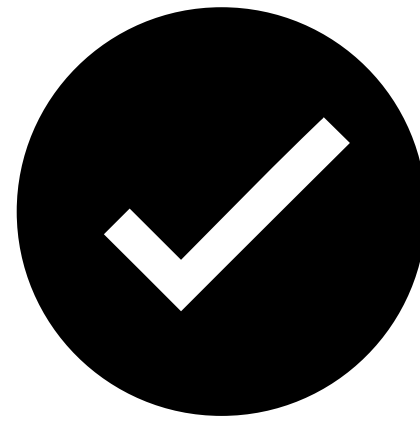


Work break-down Table :

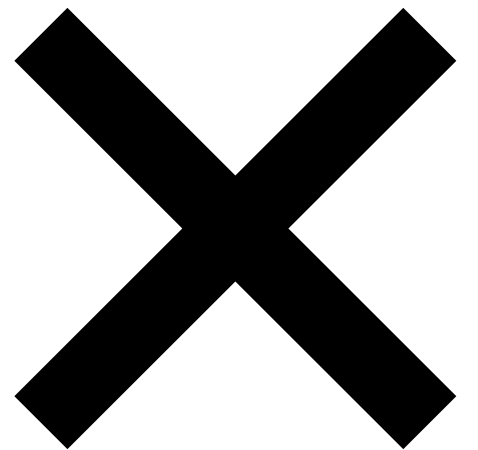
Task	Programmer Name	Programmer ID
<i>Comparison operators < , ==</i>	Ziad Tawfik Abdelnabi	20221066
<i>Plus Operator And '>'</i>	Mostafa Mohamed Anwar	20221153
<i>Minus Operator</i>	Osama Refaat Sayed	20221015
<i>Insertion operator</i>	Mostafa Mohamed Anwar	20221153
<i>Constructors Header File Invalid Function</i>	Mostafa Mohamed Anwar Osama Refaat Sayed	20221153 20221015



Algorithm Explanation (Pseudocode)



Invalid Function



Invalid function

Bool function isValid(s: string)

ctr = 0

pos = false

ng = false

ind_p = 0

ind_n = 0

for i in range(0, s.length())

if isAlpha(s[i])

return false

if s[i] == '.'

ctr++

if s[i] == '+'

pos = true

ind_p = i

if s[i] == '-'

ng = true

ind_n = i

if s[i] == ' '

return false

if ng AND pos OR s.length() == 0

return false

else if pos

if ind_p

return false

else

return true

else if ng

if ind_n

return false

else

return true

if ctr > 1

return false

else

return true



Operator Greater than



> operator

```
Bool function operator>(a: BigReal)
  if NOT isValid(number) OR NOT isValid(a.number)
    not_valid()
    exit(0)

  neg1 = a.neg
  if neg AND NOT neg1
    return false
  else if NOT neg AND neg1
    return true
  else if NOT neg AND NOT neg1
    ctr1 = 0
    ctr2 = 0
    ind1 = 0
    ind2 = 0

    for i in range(0, number.length())
      if number[i] == '.'
        ind1 = i
        break
      else
        ctr1++

    for i in range(0, a.number.length())
      if a.number[i] == '.'
        ind2 = i
        break
      else
        ctr2++
```

```
  if ctr1 > ctr2
    return true
  else if ctr1 < ctr2
    return false
  else
    for i in range(0, ind1)
      if (number[i] - '0') > (a.number[i] - '0')
        return true
      else if (number[i] - '0') < (a.number[i] - '0')
        return false

    diff1 = number.length() - ind1
    diff2 = a.number.length() - ind2

    if diff1 <= diff2
      for i in range(ind1, number.length())
        if (number[i] - '0') > (a.number[i] - '0')
          return true
        else if (number[i] - '0') < (a.number[i] - '0')
          return false
      return false
    else
      k = 0
      for i in range(ind2, a.number.length())
        if (number[i] - '0') > (a.number[i] - '0')
          return true
        else if (number[i] - '0') < (a.number[i] - '0')
          return false
      k = i

      for i in range(k + 1, number.length())
        if (number[i] - '0' > 0)
          return true
        return false
```

else

temp1 = ""

temp2 = ""

for i in range(1, number.length())

temp1 += number[i]

for i in range(1, a.number.length())

temp2 += a.number[i]


b = BigReal(temp1)

c = BigReal(temp2)


if b > c OR b == c

return false

return true



Operator less than
Operator ==



< operator

Bool function operator<(a: BigReal)

if NOT isValid(number) OR NOT isValid(a.number)

not_valid()

exit(0)

if operator>(a)

return false

if operator==(a)

return false

return true

== Operator

Bool function operator==(a: BigReal)

if NOT isValid(number) OR NOT isValid(a.number)

not_valid()

exit(0)

v = min(number.length(), a.number.length())

for i in range(0, v)

if number[i] != a.number[i]

return false

if operator>(a) OR operator<(a)

return false

return true



Plus Operator



Plus Operator

BigReal function operator+(a: BigReal)

if NOT isValid(number) OR NOT isValid(a.number)

not_valid()

exit(0)

c = BigReal()

if neg AND a.neg

temp1 = ""

temp2 = ""

for i in range(1, number.length())

temp1 += number[i]

for i in range(1, a.number.length())

temp2 += a.number[i]

b = BigReal(temp1)
d = BigReal(temp2)

c = b + d

c.number = "-" + c.number

return c

ind1 = 0

ind2 = 0

for i in range(0, number.length())

if number[i] == '.'

ind1 = i

break

for i in range(0, a.number.length())

if a.number[i] == '.'

ind2 = i

break

if NOT ind1 AND ind2

ok = false

car = 0

```

if ind2 <= number.length()
    for i in range(a.number.length() - 1, ind2 - 1, -1)
        c.number += a.number[i]

n_num = "0" * (number.length() - ind2)

for i in range(0, ind2)
    n_num += a.number[i]

i = number.length() - 1
while i >= 0
    sum = (number[i] - '0') + (n_num[i] - '0')
    sum += car

    if sum < 10
        c.number += (sum + '0')
        car = 0
    else
        ch = to_string(sum)
        c.number += ch[1]
        car = 1
    i--

```

```

if car
    c.number += (car + '0')

reverse(c.number.begin(), c.number.end())

return c

if ind2 AND NOT ind1
    c = a + *this

return c

if NOT ind1 AND NOT ind2
    if number.length() >= a.number.length()
        n_num = "0" * (number.length() - a.number.length())
        for i in range(0, a.number.length())
            n_num += a.number[i]
        car = 0
        for i in range(number.length() - 1, -1, -1)
            sum = (n_num[i] - '0') + (number[i] - '0')
            sum += car

            if sum < 10
                c.number += (sum + '0')
                car = 0
            else
                ch = to_string(sum)
                c.number += ch[1]
                car = 1

    if car
        c.number += '1'

```

```

while c.number.length() > 1 AND c.number[0] == '0' AND c.number[1] != '.'
    c.number = c.number.substr(1)

    if inte
        c.number.erase(c.number.end() - 1)

reverse(c.number.begin(), c.number.end())

return c

if number.length() < a.number.length()
    c = a + *this

return c

if ind1 >= ind2
    h = abs(ind1 - ind2)

    n_num = "0" * h
    for i in range(0, a.number.length())
        n_num += a.number[i]

    car = 0
    k = number.length()

```

```

if number.length() > n_num.length()
    for i in range(number.length() - 1, n_num.length() - 1, -1)
        c.number += number[i]
        k = i

if n_num.length() > number.length()
    for i in range(n_num.length() - 1, number.length() - 1, -1)
        c.number += n_num[i]
        k = i

i = k - 1
while i >= 0
    if number[i] == '.'
        c.number += '.'
        continue

    sum = (number[i] - '0') + (n_num[i] - '0')
    sum += car

    if sum < 10
        c.number += (sum + '0')
        car = 0
    else
        ch = to_string(sum)
        c.number += ch[1]
        car = 1

i—

```



```
if car
```

```
    c.number += (car + '0')
```

```
    reverse(c.number.begin(), c.number.end())
```

```
    return c
```

```
c = a + *this
```

```
return c
```



Minus Operator



Minus operator

BigReal function operator- (a: BigReal)

neg1 = a.neg

c = BigReal()

if neg AND NOT neg1

temp1 = ""

temp2 = ""

for i in range(1, number.length())

temp1 += number[i]

for i in range(1, a.number.length())

temp2 += a.number[i]

b = BigReal(temp1)

d = BigReal(temp2)

c = b - d

c.number = '-' + c.number

return c

if neg AND NOT neg1

number.erase(number.begin())

neg = false

c = a - *this

number = "-" + number

neg = true

return c

if NOT neg AND neg1

a.number.erase(a.number.begin())

a.neg = false

c = *this + a

a.number = "-" + a.number

a.neg = true

return c

if neg AND neg1

a.number.erase(a.number.begin())

a.neg = false

number.erase(number.begin())

neg = false

c = a - *this

number = "-" + number

neg = true

a.number = "-" + a.number

a.neg = true

return c



Machine language Simulator

0000000	0000	0001	0001	1010	0010	0001	0004	0128
0000010	0000	0016	0000	0028	0000	0010	0000	0020
0000020	0000	0001	0004	0000	0000	0000	0000	0000
0000030	0000	0000	0000	0010	0000	0000	0000	0204
0000040	0004	8384	0084	c7c8	00c8	4748	0048	e8e9
0000050	00e9	6a69	0069	a8a9	00a9	2828	0028	fdfc
0000060	00fc	1819	0019	9898	0098	d9d8	00d8	5857
0000070	0057	7b7a	007a	bab9	00b9	3a3c	003c	8888
0000080	8888	8888	8888	8888	288e	be88	8888	8888
0000090	3b83	5788	8888	8888	7667	778e	8828	8888

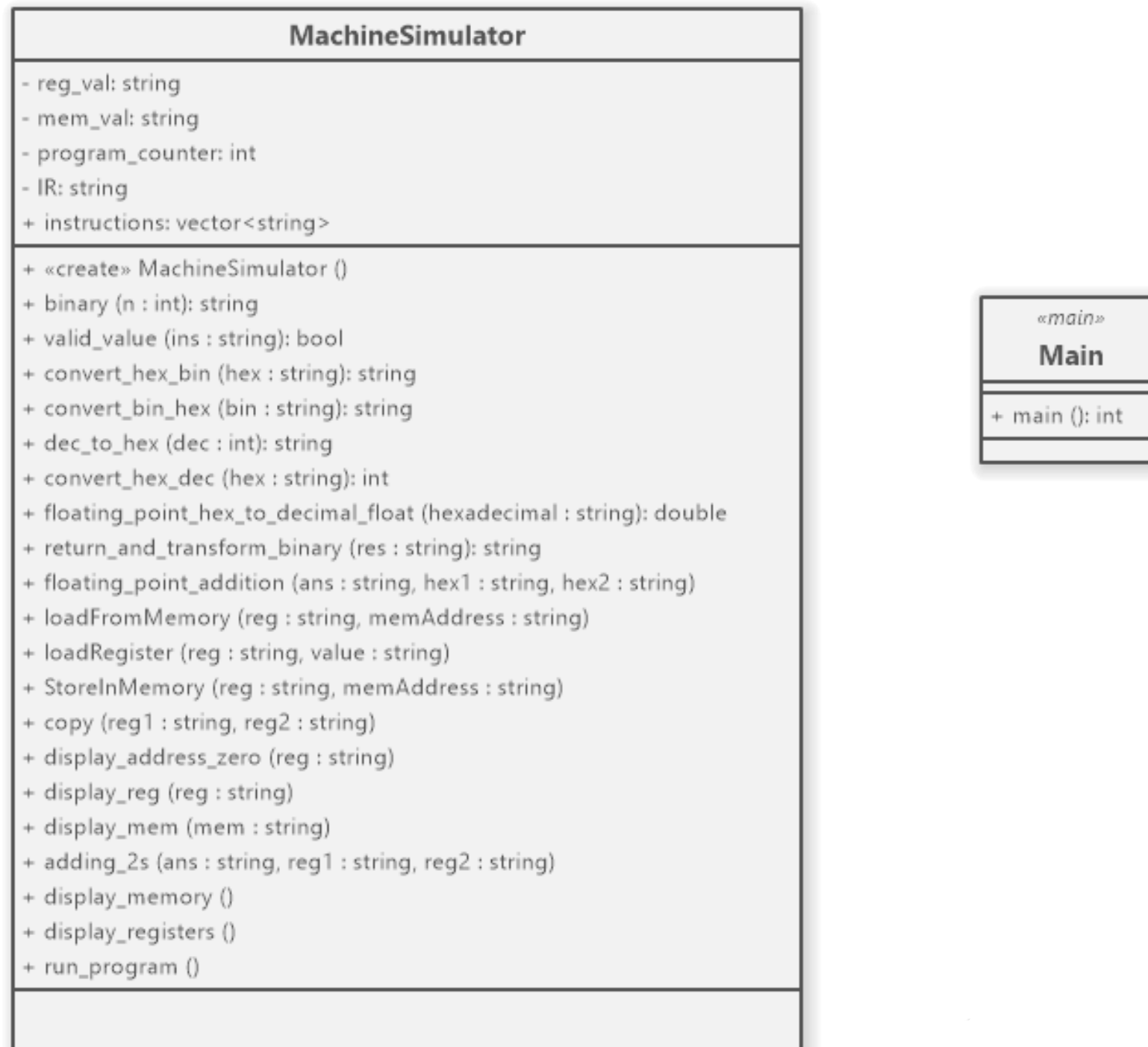


Work break-down Table :

Task	Programmer Name	Programmer ID
<i>PC-IR-FETCH-Jump</i>	Ziad Tawfik Abdelnabi	20221066
Floating point Addition	Osama Refaat Sayed	20221015
Two's Complement Addition Valid Function	Mostafa Mohamed Anwar	20221153
Constructors Header File <i>Rest of Instructions Algorithms</i>	Mostafa Mohamed Anwar Osama Refaat Sayed	20221153 20221015

UML

CLASS DIAGRAM



Github

“With 65+ Commit”

UsamaRefaat / Assignment-2-Programming-II

Code Issues Pull requests Actions Projects Security Insights Settings

Assignment-2-Programming-II Public

main 1 branch 0 tags

Go to file Add file Code

UsamaRefaat Update machine_simulator.cpp 25c1b90 14 minutes ago 68 commits

20221015 Problems	Update A2_21_20221015_12	last week
20221066 problems	Update A2_22_20221066_8.cpp	2 weeks ago
20221153 Problems	Add files via upload	last week
Task 2	Update Bigreal.cpp	last week
Task3	Update machine_simulator.cpp	14 minutes ago
README.md	Initial commit	3 weeks ago

Update Bigreal.cpp Mostafa602 committed last week	Verified	4fc6cb6	<>
Merge pull request #11 from Mostafa602/main UsamaRefaat committed last week	Verified	0a64574	<>
Update Bigreal.cpp Mostafa602 committed last week	Verified	8e41973	<>
Merge pull request #10 from ZiadTawfik2003/main UsamaRefaat committed last week	Verified	986ca26	<>
Update and rename A2.cpp to main.cpp UsamaRefaat committed last week	Verified	992f6a6	<>
Update Bigreal.cpp UsamaRefaat committed last week	Verified	5a74314	<>
Update Bigreal.cpp UsamaRefaat committed 2 weeks ago	Verified	a9111fe	<>
Update A2_22_20221066_8.cpp ZiadTawfik2003 committed 2 weeks ago	Verified	5d4735f	<>
Update A2_22_20221066_5.cpp ZiadTawfik2003 committed 2 weeks ago	Verified	4258da0	<>

Thank You!

