Đã bắt đầu vào lúc	Thứ sáu, 15 Tháng chín 2023, 4:55 PM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Chủ nhật, 17 Tháng chín 2023, 2:09 PM
Thời gian thực hiện	1 ngày 21 giờ
Điểm	11,00/11,00
Điểm	**10,00** của 10,00 (**100**%)

Câu hỏi 1

Chính xác

Điểm 1,00 của 1,00

Hoang is a K19 student studying at Bach Khoa University. He plans to write a book management software for the library. In the class design, Hoang has designed the class Book as follows:

```
class Book
{
private:
    char* title;
    char* authors;
    int publishingYear;
public:
   // some method
}
```

Your task in this exercise is to implement functions marked with /* * STUDENT ANSWER */.

Note: For exercises in Week 2, we have #include <bits/stdc++.h> and using namespace std;

For example:

Test	Result

`Book book1("Giai tich 1","Nguyen Dinh Huy",2000);`
`book1.printBook();`
	`Giai tich 1`
	`Nguyen Dinh Huy`
	`2000`

`Book book1("Giai tich 1","Nguyen Dinh Huy",2000);`
`Book book2 = book1;`
`book2.printBook();`
	`Giai tich 1`
	`Nguyen Dinh Huy`
	`2000`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
1  class Book
2  {
3  private:
4      char* title;
5      char* authors;
6      int publishingYear;
```

```
7
8  public:
9      Book()
10     {
11         /*
12          * STUDENT ANSWER
13          * TODO: set zero publishingYear and null pointer
14          */
15         title = nullptr;
16         authors = nullptr;
17         publishingYear = 0;
18     }
19
20     Book(const char* title, const char* authors, int publishingYear)
21     {
22         /*
23          * STUDENT ANSWER
24          * TODO: implement constructor to initialize title, authors, and publish
25          */
26         this->title = new char[strlen(title) + 1];
27         strcpy(this->title, title);
28
29         this->authors = new char[strlen(authors) + 1];
30         strcpy(this->authors, authors);
31
32         this->publishingYear = publishingYear;
33     }
34
35     Book(const Book &book)
36     {
37         /*
38          * STUDENT ANSWER
39          * TODO: deep copy constructor
40          */
41         this->title = new char[strlen(book.title) + 1];
42         strcpy(this->title, book.title);
43
44         this->authors = new char[strlen(book.authors) + 1];
45         strcpy(this->authors, book.authors);
46
47         this->publishingYear = book.publishingYear;
48     }
```

```
49
50      void setTitle(const char* title)
51      {
52          /*
53           * STUDENT ANSWER
54           * TODO: implement setter for title
55           */
56          delete[] this->title;
57          this->title = new char[strlen(title) + 1];
58          strcpy(this->title, title);
59      }
60
61      void setAuthors(const char* authors)
62      {
63          /*
64           * STUDENT ANSWER
65           * TODO: implement setter for authors
66           */
67          delete[] this->authors;
68          this->authors = new char[strlen(authors) + 1];
69          strcpy(this->authors, authors);
70      }
71
72      void setPublishingYear(int publishingYear)
73      {
74          /*
75           * STUDENT ANSWER
76           * TODO: implement setter for publishingYear
77           */
78          this->publishingYear = publishingYear;
79      }
80
81      const char* getTitle() const
82      {
83          /*
84           * STUDENT ANSWER
85           * TODO: implement getter for title
86           */
87          return title;
88      }
89
```

```
90      const char* getAuthors() const
91      {
92          /*
93           * STUDENT ANSWER
94           * TODO: implement getter for authors
95           */
96          return authors;
97      }
98
99      int getPublishingYear() const
100     {
101         /*
102          * STUDENT ANSWER
103          * TODO: implement getter for publishingYear
104          */
105         return publishingYear;
106     }
107
108     ~Book()
109     {
110         /*
111          * STUDENT ANSWER
112          * TODO: implement destructor to free memory
113          */
114         delete[] title;
115         delete[] authors;
116     }
117
118     void printBook(){
119         printf("%s\n%s\n%d", this->title, this->authors, this->publishingYear);
120     }
121 };
122
```

	Test	Expected	Got	

✔	`Book book1("Giai tich 1","Nguyen Dinh Huy",2000);`
	`book1.printBook();`
		`Giai tich 1`
		`Nguyen Dinh Huy`
		`2000`
			`Giai tich 1`
			`Nguyen Dinh Huy`
			`2000`
				✔

✔	`Book book1("Giai tich 1","Nguyen Dinh Huy",2000);`
	`Book book2 = book1;`
	`book2.printBook();`
		`Giai tich 1`
		`Nguyen Dinh Huy`
		`2000`
			`Giai tich 1`
			`Nguyen Dinh Huy`
			`2000`
				✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

In this exercise, you can use implemented functions in *previous question* (if needed) and implement these following functions.

```
friend bool checkAuthor(Book book, char* author){}
```

In the authors attribute, it is possible to have more than one author writing a book together. So authors will have the following format: "author1, author2, ..., authorN"

The function returns true if the author is on the book's authors list, otherwise it returns false

Note: Both first and last name must match. If only a partial match, the function still returns false

For example:

Test	Result
`Book book1("Giai tich 1","Nguyen Dinh Huy, Nguyen Thi Xuan Anh",2000);` `cout << checkAuthor(book1,"Nguyen Dinh Huy");`	1
`Book book1("Giai tich 1","Nguyen Dinh Huy, Nguyen Thi Xuan Anh",2000);` `cout << checkAuthor(book1,"Nguyen Thi Xuan");`	0

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
class Book
{
private:
    char *title;
    char *authors;
    int publishingYear;

public:
    Book()
    {
        /*
         * STUDENT ANSWER
         * TODO: set zero publishingYear and null pointer
         */
```

```
        title = nullptr;
        authors = nullptr;
        publishingYear = 0;
    }

    Book(const char* title, const char* authors, int publishingYear)
    {
        /*
         * STUDENT ANSWER
         * TODO: implement constructor to initialize title, authors, and publish
         */
        this->title = new char[strlen(title) + 1];
        strcpy(this->title, title);

        this->authors = new char[strlen(authors) + 1];
        strcpy(this->authors, authors);

        this->publishingYear = publishingYear;
    }

    Book(const Book &book)
    {
        /*
         * STUDENT ANSWER
         * TODO: deep copy constructor
         */
        this->title = new char[strlen(book.title) + 1];
        strcpy(this->title, book.title);

        this->authors = new char[strlen(book.authors) + 1];
        strcpy(this->authors, book.authors);

        this->publishingYear = book.publishingYear;
    }

    void setTitle(const char* title)
    {
        /*
         * STUDENT ANSWER
         * TODO: implement setter for title
         */
        delete[] this->title;
```

```
57          this->title = new char[strlen(title) + 1];
58          strcpy(this->title, title);
59      }
60
61      void setAuthors(const char* authors)
62      {
63          /*
64           * STUDENT ANSWER
65           * TODO: implement setter for authors
66           */
67          delete[] this->authors;
68          this->authors = new char[strlen(authors) + 1];
69          strcpy(this->authors, authors);
70      }
71
72      void setPublishingYear(int publishingYear)
73      {
74          /*
75           * STUDENT ANSWER
76           * TODO: implement setter for publishingYear
77           */
78          this->publishingYear = publishingYear;
79      }
80
81      const char* getTitle() const
82      {
83          /*
84           * STUDENT ANSWER
85           * TODO: implement getter for title
86           */
87          return title;
88      }
89
90      const char* getAuthors() const
91      {
92          /*
93           * STUDENT ANSWER
94           * TODO: implement getter for authors
95           */
96          return authors;
97      }
98
```

```
    int getPublishingYear() const
    {
        /*
         * STUDENT ANSWER
         * TODO: implement getter for publishingYear
         */
        return publishingYear;
    }

    ~Book()
    {
        /*
         * STUDENT ANSWER
         * TODO: implement destructor to free memory
         */
        delete[] title;
        delete[] authors;
    }

    friend bool checkAuthor(Book book, const char* author)
    {
        string res = "";
    for (int i = 0; i < strlen(author); i++){
        res += author[i];
    }
    string s = "";
    for (int i = 0; i < strlen(book.authors); i++){
        s += book.authors[i];
    }
    stringstream ss (s);
    string word;
    while (getline (ss, word, ',')) {
        if (word[0] == ' ') {
            word.erase(0, 1);

        }
        if (word == res) return true;
    }
    return false;
    }
};
```

	Test	Expected	Got	
✔	`Book book1("Giai tich 1","Nguyen Dinh Huy, Nguyen Thi Xuan Anh",2000);` `cout << checkAuthor(book1,"Nguyen Dinh Huy");`	1	1	✔
✔	`Book book1("Giai tich 1","Nguyen Dinh Huy, Nguyen Thi Xuan Anh",2000);` `cout << checkAuthor(book1,"Nguyen Thi Xuan");`	0	0	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

In this exercise, you will implement function **printBook(const Book book)** in **class Printer** to print information of the book. See example for output format (no spaces at the end of each line and no empty lines at the end).

Note: In the authors attribute, it is possible to have more than one author writing a book together. So authors will have the following format: "author1, author2, ..., authorN"

For example:

Test	Result

`Book book1("Giai tich 1", "Nguyen Dinh Huy, Nguyen Thi Xuan Anh", 2000);`
`Printer::printBook(book1);`
	`Giai tich 1`
	`Nguyen Dinh Huy`
	`Nguyen Thi Xuan Anh`
	`2000`

`Book book1("Introduction to Algorithms", "Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein", 1990);`
`Printer::printBook(book1);`
	`Introduction to Algorithms`
	`Thomas H. Cormen`
	`Charles E. Leiserson`
	`Ronald L. Rivest`
	`Clifford Stein`
	`1990`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
1  class Book
2  {
3  private:
4      char *title;
5      char *authors;
6      int publishingYear;
7
8  public:
9       Book()
```

```
    {
        /*
         * STUDENT ANSWER
         * TODO: set zero publishingYear and null pointer
         */
        title = NULL;
        authors = NULL;
        publishingYear = 0;
    }

    Book(const char* title, const char* authors, int publishingYear)
    {
        /*
         * STUDENT ANSWER
         * TODO: implement constructor to initialize title, authors, and publishi
         */
        this->title = new char[strlen(title) + 1];
        strcpy(this->title, title);

        this->authors = new char[strlen(authors) + 1];
        strcpy(this->authors, authors);

        this->publishingYear = publishingYear;
    }

    Book(const Book &book)
    {
        /*
         * STUDENT ANSWER
         * TODO: deep copy constructor
         */
        this->title = new char[strlen(book.title) + 1];
        strcpy(this->title, book.title);

        this->authors = new char[strlen(book.authors) + 1];
        strcpy(this->authors, book.authors);

        this->publishingYear = book.publishingYear;
    }

    ~Book()
    {
```

```
        /*
         * STUDENT ANSWER
         * TODO: implement destructor to free memory
         */
        delete[] this->title;
        delete[] this->authors;
        this->publishingYear = 0;
    }

    friend class Printer;
};

class Printer
{
public:
    static void printBook(const Book book)
    {
        for (int i = 0; i < strlen(book.title); i++){
            cout << book.title[i];
        }
        cout << endl;
        for (int i = 0; i < strlen(book.authors); i++){
            if (book.authors[i] == ',') {
                i += 2;
                cout << endl;
            }
            cout << book.authors[i];
        }
        cout << endl;
        cout << book.publishingYear;
    }
};
```

	Test	Expected	Got	
✔	`Book book1("Giai tich 1", "Nguyen Dinh Huy, Nguyen Thi Xuan Anh", 2000);` `Printer::printBook(book1);`	Giai tich 1 Nguyen Dinh Huy Nguyen Thi Xuan Anh 2000	Giai tich 1 Nguyen Dinh Huy Nguyen Thi Xuan Anh 2000	✔
✔	`Book book1("Introduction to Algorithms", "Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein", 1990);` `Printer::printBook(book1);`	Introduction to Algorithms Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein 1990	Introduction to Algorithms Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein 1990	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 4

Chính xác

Điểm 1,00 của 1,00

In a game, we have class `Character` to store characters' data.
The class `Character` is declared as below:

```
class Character {
protected:
    int hp;
    int x;
    int y;
public:
    // Constructor: set the values of x and y and hp to 0
    Character();

    // Constructor: set the values of hp, x and y to each parameter
    Character(int hp, int x, int y);

    // Set and get hp
    int getHp();
    void setHp(int hp);

    // Set and get x
    int getX();
    void setX(int x);

    // Set and get y
    int getY();
    void setY(int y);

    // Get Manhattan distance to other character
    int getManhattanDistTo(Character* other);
};
```

Your task is to define the constructors and the methods of the class.

Note:
In this task, `iostream` library has been included, and `namespace std` is being used. No other libraries are allowed.

For example:

Test	Result
`Character ch1(100, 3, 6);` `cout << ch1.getHp() << " " << ch1.getX() << " " << ch1.getY();`	`100 3 6`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
Character::Character() {
    // STUDENT ANSWER
    hp = 0;
    x= 0;
    y = 0;
}

Character::Character(int hp, int x, int y) {
    // STUDENT ANSWER
    this->hp = hp;
    this->x = x;
    this->y = y;
}

int Character::getHp() {
    // STUDENT ANSWER
    return this->hp;
}

void Character::setHp(int hp) {
    // STUDENT ANSWER
    this->hp = hp;
}

int Character::getX() {
    // STUDENT ANSWER
    return this->x;
}

void Character::setX(int x) {
    // STUDENT ANSWER
```

```
    this->x = x;
}

int Character::getY() {
    // STUDENT ANSWER
    return this->y;
}

void Character::setY(int y) {
    // STUDENT ANSWER
    this->y = y;
}

int Character::getManhattanDistTo(Character* other) {
    // STUDENT ANSWER
    int a = (other->x) - (this->x);
    if (a < 0) {
        a = -a;
    }
    int b = (other->y) - (this->y);
    if (b < 0) {
        b = -b;
    }
    return a + b;
}
```

	Test	Expected	Got	
✔	`Character ch1(100, 3, 6);` `cout << ch1.getHp() << " " << ch1.getX() << " " << ch1.getY();`	`100 3 6`	`100 3 6`	✔

	Test	Expected	Got	
✔	`Character ch2;` `cout << ch2.getHp() << " " << ch2.getX() << " " << ch2.getY();`	0 0 0	0 0 0	✔

✔	`Character* ch31 = new Character(100, 1, 2);`
	`Character* ch32 = new Character(100, -3, 4);`
	`cout << ch31->getManhattanDistTo(ch32);`
	`delete ch31;`
	`delete ch32;`
		6	6	✔

✔	`Character ch4;`
	`ch4.setX(4);`
	`cout << ch4.getX();`
		4	4	✔

✔	`Character ch5;`
	`ch5.setY(5);`
	`cout << ch5.getY();`
		5	5	✔

✔	`Character ch6;`
	`ch6.setHp(6);`
	`cout << ch6.getHp();`
		6	6	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 5

Chính xác

Điểm 1,00 của 1,00

In a game, we have class `Character` to store characters' data.
The class `Character` is declared as below:

```
class Character {
protected:
    int hp;
    int x;
    int y;
public:
    Character();
    Character(int hp, int x, int y);
    int getHp();
    void setHp(int hp);
    int getX();
    void setX(int x);
    int getY();
    void setY(int y);
    int getManhattanDistTo(Character* other);

    // Operator =: copy all data from Character other
    void operator=(const Character& other);

    // Operator <: Character a < Character b when a's hp is less than or equal b's hp
    bool operator<(const Character& other);

    // Operator () with zero parameters: print data of the instance with format: hp-x-y
    void operator()();
};
```

Your task is to overload these following operators: `=`, `<` and `()`. Their functions are described above.

Note:
In this task, `iostream` library has been included, and `namespace std` is being used. No other libraries are allowed.

For example:

Test	Result
Character ch1(100, 3, 6); ch1();	100-3-6

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
// Copy all data from Character other
void Character::operator=(const Character& other) {
    // STUDENT ANSWER
    this->hp = other.hp;
    this->x = other.x;
    this->y = other.y;
}

// Character a < Character b when a's hp is less than or equal b's hp
bool Character::operator<(const Character& other) {
    // STUDENT ANSWER
    if (this->hp <= other.hp) return true;
    return false;
}

// Print data of the instance with format: hp-x-y
void Character::operator()() {
    cout << this->hp << '-' << this->x << '-' << this->y;
}
```

	Test	Expected	Got	
✔	Character ch1(100, 3, 6); ch1();	100-3-6	100-3-6	✔

	Test	Expected	Got	

✔	`Character ch21(10, 20, 30);`
	`Character ch22(5, 5, 6);`
	`cout << ((ch21 < ch22) ? "true" : "false");`
		`false`	`false`	✔

✔	`Character ch31;`
	`Character ch32;`
	`cout << ((ch31 < ch32) ? "true" : "false");`
		`true`	`true`	✔

✔	`Character ch4;`
	`ch4();`
	`cout << "\n";`
	`ch4 = Character(5, 10, 20);`
	`ch4();`
		`0-0-0`
		`5-10-20`
			`0-0-0`
			`5-10-20`
				✔

✔	`Character(3, 4, 5)();` `cout << ((Character(3, 4, 5) < Character(3, 4, 5)) ? "true" : "false");`	`3-4-5true`	`3-4-5true`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 6

Chính xác

Điểm 1,00 của 1,00

In a game, we have class `Character` to store characters' data.
The class `Character` is declared as below:

```
class Character {
private:
    int x;
    int y;
protected:
    int hp;
public:
    Character();
    Character(int hp, int x, int y);
    int getHp();
    void setHp(int hp);
    int getX();
    void setX(int x);
    int getY();
    void setY(int y);
    int getManhattanDistTo(Character* other);
    void operator()();
};
```

Your task is to define a new class Player which is a derived class of class Character. The requirements of the new class are listed below:

- Methods of base class `Character` cannot be accessed outside `Player` class using `Player` instances
  Example: `Player pl; pl.setX();` will raise errors when compiled.
- Player class has these methods and constructors:
  - Constructor `Player()`: acts just like `Character()`
  - Constructor `Player(int hp, int x, int y)`: acts just like `Character(hp, x, y)`
  - Method `void printPlayerData()`: prints data of the instance with format: `hp-x-y`
  - Method `void moveTo(int x, int y)`: sets the values of `x`, `y` to new values
- The mentioned constructors and methods can be accessed outside `Player` class.

Note:
In this task, `iostream` library has been included, and `namespace std` is being used. No other libraries are allowed.

For example:

Test	Result
`Player pl1(100, 3, 6);` `pl1.printPlayerData();`	`100-3-6`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
class Player : private Character {
public:
    Player() : Character() {};
    Player(int hp, int x, int y) : Character(hp, x, y) {};

    void printPlayerData() {
        cout << getHp() << "-" << getX() << "-" << getY() << std::endl;
    }

    void moveTo(int newX, int newY) {
        setX(newX);
        setY(newY);
    }
};

```

	Test	Expected	Got	
✔	`Player pl1(100, 3, 6);` `pl1.printPlayerData();`	`100-3-6`	`100-3-6`	✔
✔	`Player pl2;` `pl2.printPlayerData();`	`0-0-0`	`0-0-0`	✔

✔	`Player pl3(300, 1, 2);`
	`pl3.moveTo(3, 4);`
	`pl3.printPlayerData();`
		`300-3-4`	`300-3-4`	✔

✔	`Player pl4(300, 1, 2);`
	`const bool condition = (is_unambiguous_public_base_of<Character>(&pl4) == nullptr && is_base_of<Character, Player>::value == true);`
	`assert(condition);`
				✔

✔	`Player pl5(300, 1, 2);`
	`pl5.moveTo(9, 7);`
	`pl5.printPlayerData();`
		`300-9-7`	`300-9-7`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 7

Chính xác

Điểm 1,00 của 1,00

In the coordinate plane, a circle is defined by center and radius.

Your task in this exercise is to implement functions marked with /* * STUDENT ANSWER */.

Note: you can use implemented class Point in *previous question*

For example:

Test	Result
`Circle A;` `A.printCircle();`	`Center: {0.00, 0.00} and Radius 0.00`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
class Point
{
private:
    double x, y;

public:
    Point()
    {
        x = 0;
        y = 0;
    }

    Point(double x, double y)
    {
        this->x = x;
        this->y = y;
    }

    void setX(double x)
    {
        this->x = x;
    }

```

```
24      void setY(double y)
25      {
26          this->y = y;
27      }
28
29      double getX() const
30      {
31          return this->x;
32      }
33
34      double getY() const
35      {
36          return this->y;
37      }
38
39      double distanceToPoint(const Point& pointA)
40      {
41          double a = this->x - pointA.x;
42          double b = this->y - pointA.y;
43          return sqrt((a * a + b * b));
44      }
45  };
46
47  class Circle
48  {
49  private:
50      Point center;
51      double radius;
52
53  public:
54      Circle()
55      {
56          /*
57           * STUDENT ANSWER
58           * TODO: set zero center's x-y and radius
59           */
60          center.setX(0);
61          center.setY(0);
62          radius = 0;
63      }
64
65      Circle(Point center, double radius)
```

```
    {
        this->center.setX(center.getX());
        this->center.setY(center.getY());
        this->radius = radius;
    }

    Circle(const Circle &circle)
    {
        this->center.setX(circle.center.getX());
        this->center.setY(circle.center.getY());
        this->radius = circle.radius;
    }

    void setCenter(Point point)
    {
        this->center.setX(point.getX());
        this->center.setY(point.getY());
    }

    void setRadius(double radius)
    {
        this->radius = radius;
    }

    Point getCenter() const
    {
        return this->center;
    }

    double getRadius() const
    {
        return this->radius;
    }

    void printCircle()
    {
        printf("Center: {%.2f, %.2f} and Radius %.2f\n", this->center.getX(), th
    }
};
```

	Test	Expected	Got	
✔	`Circle A;` `A.printCircle();`	`Center: {0.00, 0.00} and Radius 0.00`	`Center: {0.00, 0.00} and Radius 0.00`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 8

Chính xác

Điểm 1,00 của 1,00

In this exercise, you can use implemented functions in *previous question* (if needed) and implement these following functions.

```
bool containsPoint(const Point point){}
```

```
bool containsTriangle(const Point pointA, const Point pointB, const Point pointC){}
```

For example:

Test	Result
Point point0(0, 2); Point point1(1, 2); Circle A = Circle(point0, 2); cout << A.containsPoint(point1);	1
Point point0(0, 0); Point point1(1, 0), point2(-1, 0), point3(0, 3); Circle A = Circle(point0, 3); cout << A.containsTriangle(point1, point2, point3);	0

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
1  class Point
2  {
3  private:
4      double x, y;
5
6  public:
7      Point()
8      {
9          x = 0;
```

```
        y = 0;
    }

    Point(double x, double y)
    {
        this->x = x;
        this->y = y;
    }

    void setX(double x)
    {
        this->x = x;
    }

    void setY(double y)
    {
        this->y = y;
    }

    double getX() const
    {
        return this->x;
    }

    double getY() const
    {
        return this->y;
    }

    double distanceToPoint(const Point& pointA)
    {
        double a = this->x - pointA.x;
        double b = this->y - pointA.y;
        return sqrt((a * a + b * b));
    }
};

class Circle
{
private:
    Point center;
    double radius;
```

```
52
53  public:
54      Circle()
55      {
56          /*
57           * STUDENT ANSWER
58           */
59           center.setX(0);
60           center.setY(0);
61           radius = 0;
62      }
63
64      Circle(Point center, double radius)
65      {
66          /*
67           * STUDENT ANSWER
68           */
69          this->center.setX(center.getX());
70          this->center.setY(center.getY());
71          this->radius = radius;
72      }
73
74      bool containsPoint(const Point point)
75      {
76          /*
77           * STUDENT ANSWER
78           * TODO: check if a given point is entirely within the circle (does not
79                   If contain, return true.
80           */
81           double distane = sqrt(((point.getX() - this->center.getX()) * (point.ge
82           if (distane < (this->radius)) return true;
83           return false;
84      }
85
86      bool containsTriangle(const Point pointA, const Point pointB, const Point pc
87      {
88          /*
89           * STUDENT ANSWER
90           * TODO: check if a given triangle ABC (A, B, C are not on the same line
91                   If contain, return true.
92           */
93           bool ok1, ok2, ok3;
```

```
94          double distaneA = sqrt(((pointA.getX() - this->center.getX()) * (pointA
95          if (distaneA < (this->radius)) ok1 = true;
96          else ok1 = false;
97          double distaneB = sqrt(((pointB.getX() - this->center.getX()) * (pointB
98          if (distaneB < (this->radius)) ok2 = true;
99          else ok2 = false;
100         double distaneC= sqrt(((pointC.getX() - this->center.getX()) * (pointC.
101         if (distaneC < (this->radius)) ok3 = true;
102         else ok3 = false;
103         return ok1 && ok2 && ok3;
104     }
105 };
```

	Test	Expected	Got	

✔	`Point pointO(0, 2);`
	`Point point1(1, 2);`
	`Circle A = Circle(pointO, 2);`
	`cout << A.containsPoint(point1);`
		1	1	✔

✔	`Point pointO(0, 0);`
	`Point point1(1, 0), point2(-1, 0), point3(0, 3);`
	`Circle A = Circle(pointO, 3);`
	`cout << A.containsTriangle(point1, point2, point3);`
		0	0	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 9

Chính xác

Điểm 1,00 của 1,00

In this exercise, you can use implemented functions in *previous question* (if needed) and implement these following functions.

1. Overload operator =
2. Overload operator == (The two circles are equal if they have the same center and radius)
3. Overload operator >> (stdin center.x, center.y, radius in order)

For example:

Test	Input	Result

`Point point0(0, 0);`
`Circle A = Circle(point0, 3);`
`Circle B;`
`B = A;`
`cout << (B == A);`
		`1`

`Circle A;`
`cin >> A;`
`A.printCircle();`
	`2`
	`3.5`
	`2`
		`Center: {2.00, 3.50} and Radius 2.00`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
1  class Point
2  {
3  private:
4      double x, y;
5
6  public:
7      Point()
8      {
9          x = 0;
10         y = 0;
11     }
12
```

```
    Point(double x, double y)
    {
        this->x = x;
        this->y = y;
    }

    void setX(double x)
    {
        this->x = x;
    }

    void setY(double y)
    {
        this->y = y;
    }

    double getX() const
    {
        return this->x;
    }

    double getY() const
    {
        return this->y;
    }

    double distanceToPoint(const Point& pointA)
    {
        double a = this->x - pointA.x;
        double b = this->y - pointA.y;
        return sqrt((a * a + b * b));
    }
};

class Circle
{
private:
    Point center;
    double radius;

public:
```

```
    Circle()
    {
        center.setX(0);
         center.setY(0);
         radius = 0;
    }

    Circle(Point center, double radius)
    {
        this->center.setX(center.getX());
        this->center.setY(center.getY());
        this->radius = radius;
    }

    void operator=(const Circle &circle)
    {
        this->center.setX(circle.center.getX());
        this->center.setY(circle.center.getY());
        this->radius = circle.radius;
    }

    bool operator==(const Circle &circle)
    {
        return (
            (this->center.getX() == circle.center.getX()) &&
            (this->center.getY() == circle.center.getY()) &&
            (this->radius == circle.radius)
            );
    }

    friend std::istream& operator >> (std::istream &in, Circle &circle)
    {
        double a, b, c;
        in >> a >> b >> c;
        circle.center.setX(a);
        circle.center.setY(b);
        circle.radius = c;
        return in;
    }

    void printCircle()
    {
```

```
96          printf("Center: {%.2f, %.2f} and Radius %.2f\n", this->center.getX(), thi
97      }
98  };
```

	Test	Input	Expected	Got	

✔	`Point point0(0, 0);`
	`Circle A =`
	`Circle(point0, 3);`
	`Circle B;`
	`B = A;`
	`cout << (B == A);`
			1	1	✔

✔	`Circle A;`
	`cin >> A;`
	`A.printCircle();`
		2
		3.5
		2
			Center: {2.00, 3.50} and Radius 2.00	Center: {2.00, 3.50} and Radius 2.00	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 10

Chính xác

Điểm 1,00 của 1,00

In the coordinate plane, we have class Point to store a point with it's x-y coordinate.

Your task in this exercise is to implement functions marked with /* * STUDENT ANSWER */.

Note: For exercises in Week 1, we have #include <bits/stdc++.h> and using namespace std;

For example:

Test	Result
`Point A(2, 3);` `cout << A.getX() << " " << A.getY();`	2 3

`Point A(2, 3);`
`Point B(1, 1);`
`cout << pow(A.distanceToPoint(B), 2);`
	5

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
class Point
{
private:
    double x, y;

public:
    Point()
    {
        x= 0;
        y = 0;
    }

    Point(double x, double y)
    {
        this->x = x;
        this->y = y;
    }

    void setX(double x)
```

```
21          this->x = x;
22      }
23
24      void setY(double y)
25      {
26          this->y = y;
27      }
28
29      double getX() const
30      {
31          return this->x;
32      }
33
34      double getY() const
35      {
36          return this->y;
37      }
38
39      double distanceToPoint(const Point& pointA)
40      {
41          double a = this->x - pointA.x;
42          double b = this->y - pointA.y;
43          return sqrt((a * a + b * b));
44      }
45  };
```

	Test	Expected	Got	
✔	`Point A(2, 3);` `cout << A.getX() << " " << A.getY();`	2 3	2 3	✔

✔	`Point A(2, 3);`
	`Point B(1, 1);`
	`cout << pow(A.distanceToPoint(B), 2);`
		5	5	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 11

Chính xác

Điểm 1,00 của 1,00

1. In the toy store, all toy has a price. Car toy has a price and color, Puzzle toy has a price and size. We have to implement class CarToy and class PuzzleToy which inherit from class Toy.
2. class ToyBox has a pointer array to store a list of toys (up to 5 items including car and puzzle) and number of items in the box.

Your task is to implement two function addItem(...) in class ToyBox. If successfully added, the function returns the current number of toys in the box. If the box is full, return -1.

For example:

Test	Result

`CarToy car(20000,red);`
`PuzzleToy puzzle(30000,small);`
`car.printType();`
`puzzle.printType();`
	`This is a car toy`
	`This is a puzzle toy`

`CarToy car(20000,red);`
`PuzzleToy puzzle(30000,small);`
 	
`ToyBox box;`
`box.addItem(car);`
`box.addItem(puzzle);`
`box.printBox();`
	`This is a car toy`
	`This is a puzzle toy`

`Toy* toy = new CarToy(30000,red);` `toy->printType();`	`This is a car toy`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
enum Color
{
    red,
    green,
```

```
    blue
};
enum Size
{
    small,
    medium,
    big
};

class Toy
{
protected:
    double price;

public:
    Toy(double price)
    {
        this->price = price;
    }

    virtual void printType() = 0;
    friend class ToyBox;
};

class CarToy : public Toy
{
private:
    Color color;

public:
    CarToy(double price, Color color) : Toy(price)
    {
        /*
         * STUDENT ANSWER
         */
        this->color = color;
    }

    void printType()
    {
        cout << "This is a car toy\n";
    }
```

```
47
48      friend class ToyBox;
49  };
50
51  class PuzzleToy : public Toy
52  {
53  private:
54      Size size;
55
56  public:
57      PuzzleToy(double price, Size size) : Toy(price)
58      {
59          /*
60           * STUDENT ANSWER
61           */
62          this->size = size;
63      }
64
65      void printType()
66      {
67          cout << "This is a puzzle toy\n";
68      }
69
70      friend class ToyBox;
71  };
72
73  class ToyBox
74  {
75  private:
76      Toy *toyBox[5];
77      int numberOfItems;
78
79  public:
80      ToyBox()
81      {
82          /*
83           * STUDENT ANSWER
84           * TODO: set zero numberOfItems and nullptr toyBox
85           */
86          for (int i = 0; i < 5; i++){
87              toyBox[i] = NULL;
88          }
```

```
89          numberOfItems = 0;
90      }
91
92      int addItem(const CarToy &carToy)
93      {
94          /*
95           * STUDENT ANSWER
96           * TODO: function add a new Car toy to the box.
97                   If successfully added, the function returns the current number
98                   If the box is full, return -1.
99           */
100         if (numberOfItems < 5) {
101             toyBox[numberOfItems] = new CarToy (carToy);
102             numberOfItems++;
103             return numberOfItems;
104         }
105         else return -1;
106     }
107
108     int addItem(const PuzzleToy &puzzleToy)
109     {
110         /*
111          * STUDENT ANSWER
112          * TODO: function add a new Puzzle toy to the box.
113                  If successfully added, the function returns the current number
114                  If the box is full, return -1.
115          */
116         if (numberOfItems < 5) {
117             toyBox[numberOfItems] = new PuzzleToy (puzzleToy);
118             numberOfItems++;
119             return numberOfItems;
120         }
121         else return -1;
122     }
123
124     void printBox()
125     {
126         for (int i = 0; i < numberOfItems; i++)
127             toyBox[i]->printType();
128     }
129 };
```

	Test	Expected	Got	

✔	`CarToy car(20000,red);`
	`PuzzleToy puzzle(30000,small);`
	`car.printType();`
	`puzzle.printType();`
		`This is a car toy`
		`This is a puzzle toy`
			`This is a car toy`
			`This is a puzzle toy`
				✔

✔	`CarToy car(20000,red);`
	`PuzzleToy puzzle(30000,small);`
	
	`ToyBox box;`
	`box.addItem(car);`
	`box.addItem(puzzle);`
	`box.printBox();`
		`This is a car toy`
		`This is a puzzle toy`
			`This is a car toy`
			`This is a puzzle toy`
				✔

✔	`Toy* toy = new CarToy(30000,red);` `toy->printType();`	`This is a car toy`	`This is a car toy`	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.