

نام:

نام خانوادگی:

شماره دانشجویی:

گرایش تحصیلی:

بسمه تعالی  
امتحان میان ترم  
برنامه سازی پیشرفته

تاریخ : ۱۳۹۵/۰۲/۱۳

مدرس : علیرضا اخوان

زمان : ۱۱۰ دقیقه

صحيح و غلط: جملات زیر را خوانده و درستی و یا نادرستی آن ها را تعیین نمایید.

بارم

۱.

عبارت	صحيح	غلط
الف) کامپایلر برای کلاسی که تعریف میکنید یک تابع سازنده توکار (built-in) بدون آرگومان فراهم می کند، اما اگر خودتان یک تابع سازنده تعریف کنید، تابع سازنده ی توکار دیگر ایجاد نمی شود.		
ب) به صورت پیش فرض اعضای کلاس public هستند.		
ج) کلمه private و protected در کلاس باعث میشود خصیصه های آن محدوده از بیرون قابل دسترسی نباشند.		
د) تمام اعضای داخل یک union یک حافظه مشترک دارند، به اندازه جمع حافظه های اعضا.		

۴

چند گزینه ای:

- تابعی که به صورت خودکار، وقتی شیء ساخته می شود فراخوانی می شود؟ (۵/۰ نمره)

a) constant      b) constructor      c) static      d) friend

- این عملگر برای تخصیص حافظه ی پویا به کار می رود: (۵/۰ نمره)

a) new      b) delete      c) static      d) real

کد رو به رو را در نظر بگیرید و به دو سوال بعدی پاسخ دهید:

```
int a[ 7 ] = { 5, 6, 3, 8, 7, 1, 2 };
int *b;
b = a;
```

- مقدار (a+2) کدام است؟! (۵/۰ نمره)

a) آدرس به خانه ای از حافظه که در آن ۶ ذخیره شده      b) آدرس به خانه ای از حافظه که در آن ۳ ذخیره شده  
c) ۶      d) ۳

- مقدار b[3] کدام است؟ (۵/۰ نمره)

a) آدرس به خانه ای از حافظه که در آن ۷ ذخیره شده      b) آدرس به خانه ای از حافظه که در آن ۸ ذخیره شده  
c) ۷      d) ۸

- قطعه کد زیر را در نظر بگیرید، کدام یک از گزینه های زیر درست است؟ (۵/۰ نمره)

```
typedef double Real;
Real Values[5];
```

الف) یک شیء به نام values می سازد.      ب) یک آرایه ی دو بعدی از double می سازد.

ج) یک آرایه از double می سازد.      د) یک typedef double می سازد.

- اگر main به صورت (int main(int argc, char\* argv[])) نوشته شده باشد، با فراخوانی برنامه به صورت

```
my.exe salam bye 12 5
```

مقدار argc برابر میشود با: (۵/۰ نمره)

a) ۳      b) ۴  
c) ۵      d) ۶

- قطعه کد زیر را در نظر بگیرید، در انتها چه عددی در خروجی چاپ می شود؟ (۱ نمره)

<pre>void f(int x, int&amp; y, int*z) {     x = 100, y = 200, *z= 300; } int main() {     int a = 1, b=2, c=3;     f(a, b, &amp;c);     cout &lt;&lt; a+b+c &lt;&lt; endl;     return 0; }</pre>	<p>الف) ۶</p> <p>ب) ۶۰۰</p> <p>ج) ۳۰۳</p> <p>د) ۵۰۱</p> <p>ه) ۴۰۲</p> <p>و) ۲۰۴</p> <p>ز) ۴۰۲</p>
--	---

۱	<p>۳. در تعریف کلاس‌ها، متغیرهای عضو آن‌ها را معمولاً به شکل خصوصی تعریف می‌کنیم. دو دلیل برای این کار را ذکر کنید.</p>	۳
۵	<p>هدف این سوال این است که سابقه‌ی بازی اعضای تیم ملی فوتبال را ذخیره کنیم و مدت عضویت هر بازیکن در تیم ملی را به دست آوریم.</p> <p>یک بازیکن ممکن است در بازه‌های تاریخی متعددی عضو تیم ملی شود، مثلاً در دوبازه، از تاریخ ۸۷/۲/۲۳ تا ۸۸/۴/۱۲ و از تاریخ ۸۸/۱۲/۱ تا ۸۹/۸/۱۴.</p> <p>هدف این است که تعداد کل روزهایی که یک بازیکن عضو تیم ملی بوده را به دست بیاوریم.</p> <p>به این منظور کلاسی به نام <code>player</code> تعریف کنید که نام بازیکن را به همراه بازه‌های تاریخی عضویت او را ذخیره می‌کند. علاوه بر تعریف سازنده‌ی مناسب، دو متد به اهداف زیر برای این کلاس تعریف کنید:</p> <ul style="list-style-type: none"> <li>• متدی که بتوان با آن یک بازه‌ی جدید را به سوابق بازیکن اضافه کرد.</li> <li>• متدی که تعداد کل روزهای عضویت بازیکن را برگرداند.</li> </ul> <p>برای ساده شدن کار، فرض کنید کلاس <code>Date</code> به شکل زیر داده شده است. معنای عملیات تعریف شده از روی نام آن‌ها واضح است. نحوه‌ی عمل تابع <code>compare_to</code> به این ترتیب است که <code>d1.compare_to(d2)</code> مقداری منفی، مثبت یا صفر بر میگرداند. به ترتیب اگر تاریخ <code>d1</code> قبل از <code>d2</code>، بعد از <code>d2</code>، یا مساوی آن باشد.</p> <p>فرض کنید پیاده سازی بخش خصوصی و بدنه‌ی متدهای این کلاس به شکل درستی انجام شده است.</p> <pre>class Date { public:     Date(int d, int m, int y);     int get_day();     int get_month();     int get_year();     void add_one_day();     int compare_to(Date); private:     // be maa rabti nadaarad che-joori piaade-saazi shode! };</pre> <p>کلاس <code>player</code> و پیاده سازی کامل آن را بنویسید.</p> <p>در صورت تمایل میتوانید کلاسهای دیگری نیز تعریف کرده از آنها استفاده کنید.</p>	۴
۲	<pre>class Rectangle {     int width, height; public:     Rectangle(int x, int y) : width(x), height(y) {}     int area(void) { return width * height; }     int compare(Rectangle r); };</pre> <p>یک کلاس مستطیل به شکل بالانوشته شده است، یک کلاس مکعب مربع با تابع سازنده <code>y</code> یک ورودی بنویسید که تابع عضوی برای چاپ حجم آن داشته باشد، در تعریف این کلاس از کلاس مستطیل استفاده کنید.</p>	۵
۱	<p>تابع <code>compare</code> کلاس <code>Rectangle</code> سوال قبل را تعریف کنید. به نحوی که اگر در <code>main</code> برنامه کد زیر را بنویسیم، مطابق زیر مساحت اشیاء از نوع <code>Rectangle</code> را مقایسه کند.</p> <pre>if (rect1.compare(rect2))     cout &lt;&lt; " rect2 is smaller than rect1" &lt;&lt; endl; else     cout &lt;&lt; " rect2 is equal to or larger than rect1" &lt;&lt; endl;</pre>	۶

```

۲ class Complex {
public:
    Complex(double r, double i) : real(r), imag(i) {}
    Complex(double r) : real(r), imag(0) {}
    friend ostream& operator<<(ostream& out, const Complex& c);
private:
    double real;
    double imag;
};

```

الف) آیا در کلاس بالا میتوان بعد از ایجاد یک شیء از این کلاس؛ *real* و *imag* آن را در *main* تغییر داد؟ چرا؟ (۵، ۰ نمره)

ب) بدون هیچ تغییری در کلاس *Complex*، عملگر << را برای چاپ اشیائی از نوع این کلاس سربارگذاری کنید، به نحوی که

```

Complex a(3, 2);
cout << a << endl;

```

مقدار  $3i+2$  را روی خروجی نمایش دهد (۱ نمره)

ب) چرا در تعریف این کلاس، تابع << operator را دوست خودش تعریف کرده؟ (۵، ۰ نمره)

```

۲ struct student{
    string name;
    char id[8]; //assume that studentID is 8 digit
    float moadel;
};
vector<student> top_students (vector<student> std, int threshold )
{
    //A: your code here
}

```

الف) قطعه کد بالا را در نظر بگیرید، قسمت A را به نحوی بنویسید که تابع *top\_students* یک وکتور از *student* و یک عدد به عنوان *threshold* بگیرد، و به عنوان خروجی وکتوری از *student*هایی که معدل آنها بزرگتر از *threshold* است را برگرداند.

ب) میتوانستیم به جای *struct*، از کلاس استفاده کنیم، به نظر شما مزیت تعریف کلاس نسبت به حالت فعلی چیست؟ (۵، ۰ نمره)

کد رو به رو را در نظر بگیرید، فرض کنید کاربر در ورودی عدد ۰ وارد کند که در نهایتا به خطای در حال اجرای تقسیم بر صفر برنامه را مواجه کند، الف) در *b* خطا را پرتاب کنید و در *main* برنامه آن را بگیرید و خطای در خور به کاربر نمایش دهید.

ب) اگر خطا را *b* به *main* پرتاب کند، چه بلایی سر حافظه‌ی پویای *d* میاید؟ آیا پاک می‌شود؟ چرا؟

ج) برنامه را به نحوی درست کنید که کدی که در نهایت با اصلاح تابع *b* و *main* در قسمت الف اصلاح کردید را تغییر ندهید، اما مشکل آزادسازی حافظه در هنگام بروز خطا در تابع *a* رفع شود. (نیازی نیست الف و ج را جدا گانه جواب دهید، تنها کد نهایی حاصل از قسمت الف و ج را بنویسید.)

```

int b(int i)
{
    return (3/i);
}
float a(int i)
{
    int out;
    double* d=new double[50000];
    out=b(i);
    //do sth else
    delete[] d;
    return out;
}
int main(){
    int i;
    cin>>i;
    a(i);
    //do sth else , tons of code !!
}

```

هیچ خطی از کد را جا به جا نکنید، و منطق برنامه را عوض نکنید.

تنها مجاز به اضافه کردن بلاکهای *try* و *catch* و *throw* و کدهایی برای مدیریت خطا هستید.